

REC'D 17 MAY 2004

WIPO

PCT

P1 1167213

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office

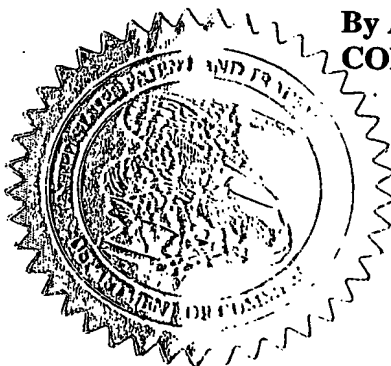
May 12, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: 60/457,277

FILING DATE: March 25, 2003

RELATED PCT APPLICATION NUMBER: PCT/US04/08681



By Authority of the
COMMISSIONER OF PATENTS AND TRADEMARKS

L. Edelen

L. EDELEN
Certifying Officer

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

DESI AVAILABLE COPY

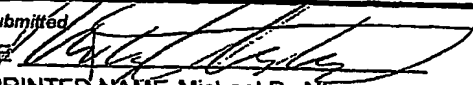
03-26-03 03/25/03

PTO/SB/16 (2-98) [Unisys Corp.]
Approved for Use through 1/31/2001. OMB 0651-0037
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

03/25/03
JC950 U.S. PTO

PROVISIONAL APPLICATION FOR PATENT COVER SHEET
This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)					
Given Name (first and middle [if any])	Family Name or Surname		Residence (City and either State or Foreign Country)		
	UNISYS CORPORATION		Blue Bell, Pennsylvania		
<input type="checkbox"/> Additional inventors are being named on the separately numbered sheets attached hereto					
TITLE OF THE INVENTION (280 characters max)					
FRONT-END (CLIENT) LINKS TO WORKFLOW ENGINE + PERMISSION SCHEMA					
CORRESPONDENCE ADDRESS					
Direct all correspondence to:		27276		Place Customer Number Bar Code Label here	
<input type="checkbox"/> Customer Number					
OR		Type Customer Number here			
<input checked="" type="checkbox"/> Firm or Individual Name		Unisys Corporation			
Address		Unisys Way			
Address		MS/E8-114			
City		Philadelphia	State	Pennsylvania	ZIP 19424
Country		USA	Telephone	(215) 986-4111	Fax (215) 986-3090
ENCLOSED APPLICATION PARTS (check all that apply)					
<input checked="" type="checkbox"/> Specification Number of Pages 271 (total, not in a single sequential order) including 1 page of Abstract and 0 pages of claims					
<input type="checkbox"/> Small Entity Statement					
<input type="checkbox"/> Drawing(s) Number of Sheets <input checked="" type="checkbox"/> Other (specify) - Drawings interspersed with text in Specification.					
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)					
<input type="checkbox"/> A check or money order is enclosed to cover the filing fees				FILING FEE AMOUNT (\$)	
<input checked="" type="checkbox"/> The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 19-3790				\$160.00	
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.					
<input checked="" type="checkbox"/> No.					
<input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are:					

Respectfully submitted
SIGNATURE 
TYPED or PRINTED NAME Michael B. Alfass
TELEPHONE (215) 986-4111

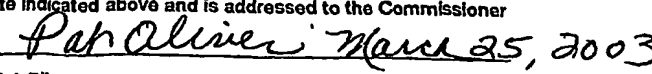
Dated: March 25, 2003

Registration No. 30,606
Docket Number: N/A

PROVISIONAL APPLICATION FILING ONLY

EXPRESS MAIL Mailing Label Number: EK719005750US Date of Deposit: March 25, 2003

I hereby certify that this paper and fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service with sufficient postage under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20231.


Pat Oliver

UNITED STATES PATENT APPLICATION**INVENTORS:****Unisys Corporation****APPLICATION:****Front-end (Client) Links to Workflow Engine + Permissions
Schema****ATTORNEY DOCKET NO.****Not yet assigned**

**Michael Atlass
Attorney for Applicant
Reg. No. 30,606
Telephone No. (215) 986-4111
Unisys Corporation
M.S. E8-114
Unisys Way
Blue Bell, PA 19424-0001**

CERTIFICATE UNDER 37 CFR 1.10: The undersigned hereby certifies that this transmittal letter and the paper of papers, as described hereinabove, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of EK719005750US, in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231 on this 25th day of March, 2003.

Pat Oliver
Pat Oliver

March 25, 2003
Date

Front-end (Client) Links to Workflow Engine + Permissions Schema

Background of the Invention

Field of the Invention:

This invention relates to the field of middleware which allows connection between clients and services on a server, and is particularly adapted to allowing non-proprietary front end clients or clients of other proprietary systems to interact efficiently with said server services.

Background:

The first application of this invention is in the field of publishing software services and systems where the company of the inventors has been active for a number of years, producing and selling a system called Hermes that supports the management of various items useful for publishing newspapers, particularly; text, pages, images, and charts. Due to the rise of small scale systems and particularly ones which use various other versions of software for working on some or all of these items, most notably Adobe InDesign and Adobe InCopy as well as QuarkExpress, many users are familiar and comfortable with using features of these programs for managing their use of these items and preparing them for publication. Therefore, even though our Hermes system has fine editors for handling such items, there has developed a need to accommodate such users.

The Hermes system maintaining a repository for manipulating, managing, sizing, layout, printing, and other related activities with respect to the items (charts, text, images, pages and the like), as do other proprietary publishing systems. Accordingly, a system and method for accomplishing such management and manipulation functions that accepts items formatted in a customer preferred front-end program is desirable.

Further, with the rise of open standards such as HTTP, XML, and SOAP, many of the characteristics of the data files for objects can be communicated about between pieces of software. However, a system has not heretofore been developed to make this all work together in a commercially viable manner.

Summary of the Invention

In order to allow the proprietary system (Hermes, in the preferred embodiment) to manage and manipulate items as objects, a set of metadata about these items is produced, identifying a set of characteristics about them which allows them to be manipulated by the proprietary system. Further, the front end or client software through which the item was created is not affected, because we provide the data file of the item in an unmodified format back to the client software should further content manipulation be needed. To accomplish this a metadata envelope is created around the item to make it into an object. The envelope is described in SOAP/XML to define and enable handling of the object and HTTP(S) is used for its transmission. Additional meta data is used to identify the native format of the item. Thus, the envelope has HTTP and XML data, as well as SOAP data to define and enable handling of the object, and a data string is also created identifying the native format of the item. Accordingly, using software such as the Woodwing SmartConnection Pro to connect to clients InDesign and InCopy, a SOAP connector is provided which forwards the partially packaged object to the workflow engine and permissions scheme (our Hermes editorial system) which completes the envelope and keeps track of what happens to the object inside of its own repository. Thus this middleware operates like a web service, generating a format field and providing HTTP/XML/SOAP connections between the workflow engine/permissions schema and the client software.

Brief Description of the Figures

Various figures are included in the accompanying documents made a part hereof by this reference thereto.

Detailed Description of the Preferred Embodiments

While various embodiments of the present invention are described, it should be understood that they have been presented by way of example only, and not as a limitation. Thus, the breadth and scope of the present invention should not be limited by

any of the above-described exemplary embodiments, but should be defined only in accordance with the Claims that may issue herefrom and their equivalents.

The first document describes the overall architecture. It is the design specification (8 pages) for the integration needed between Woodwing's SmartConnection and the Hermes SOAP Connector/Services middleware.

Database Format Field Design Specification (7 pages) is a document identifying the format field characteristics needed for the invention.

Hermes Connector Use Case Specification (23 pages) is a preferred embodiment description of use cases.

The SOAP Application Server Design Specification (12 pages) describes a preferred embodiment design of the same.

The SOAP Application Server Platform Specification (21 pages) describes a preferred embodiment design of the same.

The Third Parties Integration with NewsRoom Design Specification (10 pages) describes a preferred embodiment design of the same. NewsRoom is a component of Hermes that may have counterparts in other similar workflow and permissions systems.

The Integration Platform Project Supplementary Specification (8 pages) describes a prototype design of Hermes integration using HTTP and SOAP.

The Database Format Field Project Functional Specification (7 pages) describes a preferred embodiment design of the same.

The Adobe® Integration with eEditorial® Solutions Project Design Specification (39 pages) describes a preferred embodiment design of the same.

A 26 page chart of iconography used in the preferred embodiment is also included.

The Third Parties Integration with NewsRoom Project Functional Specification (10 pages) describes a preferred embodiment design of the same.

The Workflow Definition for External Objects Project Functional Specification (9 pages) describes a preferred embodiment design of the same.

The ADOBE InDesign Integration with News Content Manager -Hermes document (17 pages) describes a preferred embodiment design of the same.

The Advanced Query in Third Party Applications Design Specification(21 pages) describes a preferred embodiment design of the same.

The Pages and Objects Management in External Applications Project Functional Specification (12 pages) describes a preferred embodiment design of the same.

The Hermes Palette Availability in External Applications Project Functional Specification (9 pages) describes a preferred embodiment design of the same.

The Third Parties Integration with Hermes Explorer Project Functional Specification (11 pages) describes a preferred embodiment design of the same.

Thus, the invention has been described.

ABSTRACT

Middleware ties together client front-end software that creates content items such as text editor files, image files, page files, and chart files in a manner that it can be used in conjunction with a workflow and repository management system even though both systems have different proprietary formats for such items. The items are packaged as objects with an envelope to allow for protocol handling (HTTP, XML and SOAP) as well as with metadata to allow for manipulation of the objects within the workflow and repository management system. The first application of this invention is to a large scale publishing system.

UNISYS

Overall architecture

The overall architecture of the integration is depicted in figure 1

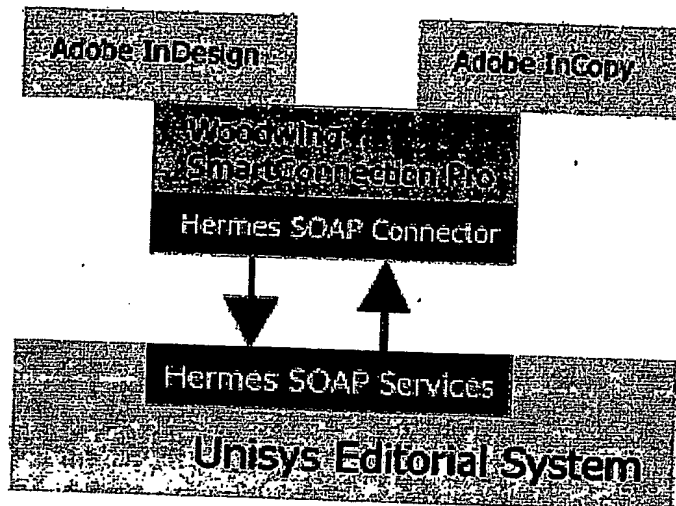


Figure 1 Overall integration architecture

The Hermes connector will provide a set of functionality that will enable the complete integration of InDesign and InCopy with Hermes.
Unisys will provide the necessary SOAP APIs to implement the requirements.

The integrated HC+SCPro suite will be available for both Windows and MacOS.

The integrated HC+SCPro suite will support multiple language resource files, in order to be available in different languages. WoodWing will provide the English version of the strings to be translated, and Unisys will provide to WoodWing translated strings: the set of languages supported will be a subset of the InDesign/InCopy supported languages.

The integrated HC+SCPro suite will have the following characteristics:

1. Hermes content storage and workflow control will be used instead of the local workstation file-system as used by standard SCPro.
2. The following SCPro features will NOT be available as part of the integration with Hermes:
 - Configuration tool
 - Ready functionality
 - Send Message.

- The SCPro palette will not update automatically; also update icons will not be available for assigned/placed articles, therefore they will not be visible.
- 3. The following additional features to SCPro v2.03 will be provided:
 - 3.1. The SCPro panel extended for the integration with Hermes will allow the end-user to delete an object/page via a push button placed in the bottom of the panel, as in all other Adobe products. The same result can be achieved via a panel menu option. This operation will update the database immediately.
 - 3.2. The client application must be able to move and/or copy InDesign pages and InCopy stories among different levels and editions. Selection of the Move Object/Page from SCPro palette, will show dialog a box to enter level, edition, object/page name and pubdate (if required) to move the asset to. The maximum visible depth of Hermes Levels should be 5. This operation will update the database immediately.
 - 3.3. Files from the file-system will be placed into InDesign documents only after they have been inserted into the Hermes database. If a local file is placed, InDesign will ask to the user to create the object in the Hermes database. The user must be able to provide the Level, Edition, pub-date if required and object name of the object being created. If the creation of the object is aborted, no files will be placed into the generated low-res is download for placement using the object ID to link the asset to the page. InDesign will disable the "edit original" function for images. If placed files are image files, appropriate OPI comments will be inserted into the EPS to reference the high-resolution path of the image on the Hermes file system. Furthermore the EPS will contain the OPI comments for the transformations applied to the image. The OPI comments will conform to both OPI standards 1.3 and 2.0.
 - 3.4. Whenever an object is placed into a page, a notification must be made to the Hermes system via the HermesLinkObject API call.
 - 3.5. Objects can be unlinked from an InDesign page directly from the query result or via a menu option. When an object is unlinked from the page, it will be removed from the page and the database will be updated immediately. This will free the object and make it available for pagination. (NOTE 1. A call to Hermes must be placed to determine if the object is locked; 2. If the document is opened at anytime after the unlink, the content will be removed resulting in the display of an empty layout).
 - 3.6. When a document is released into a specific status and the status has the property (extended status) 'READY FOR TYPESET', EPS images (one per page) will be stored as well for InDesign documents. However, to enable a "quick preview" by the rest of the Hermes client, a JPEG preview will be sent to Hermes at each save.
 - 3.7. Text Conversion - The operation is achieved by opening a Hermes textual object from the query result into the InCopy application. This will resolve in the creation of a new InCopy document which is not yet stored in the database. The text conversion rules will be defined and agreed by Unisys and WoodWing. The general scope will be limited to flat text.
 - 3.8. With the same mechanism, a Hermes textual object dropped into an InDesign page will result in the creation of a new text frame on the page not linked to Hermes.
 - 3.9. For opened documents the user must be able to change metadata such as the Author, content type, etc. The specific list of metadata is included as Appendix A. XMP technology will be used for embedding metadata into InDesign and InCopy

UNISYS

- objects. The standard Adobe UI will be extended to support all the required metadata fields.
- 3.10. Versioning. Versioning will be implemented only for objects (InCopy stories). The user can ask for an overview of old versions for opened IC documents. This will show a list of version numbers to the user. The user can select a version and open it in read-only mode. A new version of the object will be created whenever the object is released to Hermes.
 - 3.11. Event management - Status and locking change events from Hermes are applied within the Query view if applicable. Each event triggered from Hermes will have the object/page ID and the new status or lock state. InDesign and InCopy will receive events from the local client messaging agent.
 - 3.12. Messaging - InDesign and InCopy will have a shortcut to the Hermes client messaging agent in order to invoke the email and alerts user interface.
 - 3.13. Dummy Text - InCopy object in a status with an attribute different from READY FOR TYPESET will be shown as dummy text instead of the actual text, preserving the text styles and format. In order to keep the look and feel as close as possible to the actual text, a simple scrambling algorithm should be used. Initially instead of showing a dummy text, InDesign can show a placeholder or a colored empty frame in order to hide the content for the screen, EPS and JPG previews. From InCopy, the textual object cannot be opened if it lies in a status that prevents showing the content.
 - 3.14. Open on demand - InDesign and InCopy will be able to open a document by using information passed to the command line. This feature will be used to launch InDesign or InCopy from UPS Explorer in the Windows environment. If the command line implementation is determined not to be feasible, Unisys will implement the start up of InDesign/InCopy via OLE automation and WoodWing will provide an OLE automation method to be called in order to complete the object/page opening process.
 - 3.15. InDesign will produce the equivalent of Hermes logical pages while InCopy will produce the equivalent of the Hermes Object. Both of them will be stored into the Hermes repository as "external objects/pages".
4. The following SCPro features will work differently:
- 4.1. Logon/logoff functionality. The user can choose the location to login, username (all in uppercase) and password (shown as ***). Unisys will provide a common login client service that will enable only one login from different applications running on the same machine.
 - 4.2. SSL. The connection between InDesign/InCopy and Hermes SOAP will be available either as normal HTTP or HTTPS.
 - 4.3. View single object - In the layout view, InCopy will show only the story that is being edited. This functionality applies only to InCopy objects that are linked to an InDesign page. This will be implemented through the development of a function to directly zoom the editable boxes to fit the window. Non editable text will be dimmed.
 - 4.4. The Hermes connector will use it's own dialogs for Save, Assign, Check-In, Copy, and Release to replace the standard SCPro dialog.
 - The Save dialog will allow the user the select Level, Edition, Object name and publication date (if required) by the selected level.
 - The Assign dialog will allow the user to select Object name, level, edition and pubdate (if required) by the selected level. No assumption about the inheritance of

level, edition and pub-date can be made. Hermes provides different types of levels. Objects can be placed only in an object type level while pages can be placed only in page type levels. The maximum visible depth of Hermes Levels should be 5
 - The Check-in and Release dialog allows the user to select status from a list of next available statuses for the object/page.
 The client application will show a dialog box to the end-user in order to select the available statuses in which the object/page can be released. Each entry of the list of statuses will be decorated with a small square filled in with the color representing the status. Objects and pages don't have a default next status so it must be queried every time.

4.5. Instead of the hierarchical browse functionality of SCPro a Query mechanism will be used. One of the combo-boxes of the SCPro palette will be used to choose between an Object and Page query. With a palette menu the user can select the Query parameters: Publication from/to, Status, Object type (for object query), Author, Name of object (wildcards accepted). Searching the metadata will also be possible. The result of the query, returned from Hermes, is shown in the SCPro palette with the Object/Page name always visible. Other attributes such as Object Type, Object Name, Author, Status, Lock State can be displayed on demand by the user. The view can be updated with the Refresh menu option.

4.6. Selection of a New Hermes Document shows a dialog to select level, publication, grid etc. The grid definitions will be retrieved from the SOAP server. Hermes has the ability to define a page grid per level, so InDesign will query page grid names at each level selection. After the selection of a specific grid, the complete grid definition will be fetched from the server, and used to create a new InDesign document that is created on Hermes as well.

4.7. Ad spaces defined in Hermes (so called "Face off" function).
 InDesign must read Ad layout information. The layout will be defined as two pair of coordinates defining the position inside the page and the width/height of the space. The layout defining the Ad space can be moved and resized but constraint must be applied in order to ensure that the shape is always rectangular. No polygons can be created to define an Ad space.

The reference of the layout will be returned from Hermes system and must be visible inside the shape.

Ad spaces defined in InDesign

There must be the possibility to create Ad spaces in InDesign directly. In this case, the layout must have a Hermes reference defined by the user.

In both cases, the layout information will be updated when the InDesign documents is saved.

The Ad space will be visibly identified and no text or images will be placed on it. Dynamically prevent placement of objects in Ad space.

Documentation

User documentation and installation/configuration manual will be written in US-English covering the SCPro functionality in combination with the Hermes Connector, approx. 10-20 pages. Documentation will be supplied as PDF a document in letter format, and in an editable format, suitable for re-purposing as part of Unisys documentation (done in MS Word format).

Technical Design Documentation. Each Milestone Phase will begin with Woodwing providing Unisys with Technical Design Documentation suitable to understand the details of the development tasks to be undertaken. The Technical Design Documentation will consist of Use Case Diagrams, Use Case Descriptions and Sequence Diagrams. While it is probable that a single Use Case will adequately describe each the development activities, additional Use Cases, Descriptions, etc should be provided for complex subject areas. To insure consistency of diagramming and documentation style, an example of each of these artifacts is provided as Appendix B and should be used by Woodwing to guide the development of their artifacts.

Within 3 business days of receipt of the Technical Design Documentation, Unisys will respond with their approval or requested changes at which time the development work for the milestone can begin.

Additional features

Woodwing will provide a combined setup wizard, at the final stage, for both SCPro software and Hermes Connector. In addition, SCPro updates will also be delivered with an Installation wizard appropriate for the update.

Testing

WoodWing will alpha test the software.

Unisys is to provide the best effort to make available a SOAP server accessible by Woodwing in order to alpha test the software. The server will be located in Milan offices and will be accessible under strict security control and only after Unisys Corporation will confirm that all the requirements for accessing it are satisfied by Woodwing. In the event it is determined not to be technically feasible, WoodWing will be required to perform all alpha testing at the Unisys site.

Development Lifecycle Test Criteria

The parties have agreed that the acceptance of each milestone of the Software as well as the final Implementation Testing by Unisys will be governed by the following criteria.

- The Software shall contain Zero Class A TSIs (See Exhibit 4 of SOFTWARE LICENCE AND DEVELOPMENT AGREEMENT for definitions)
- The Software shall contain Zero Class B TSIs (See Exhibit 4 of SOFTWARE LICENCE AND DEVELOPMENT AGREEMENT for definitions)
- The Software shall contain a maximum of ten (10) Class C TSIs (See Exhibit 4 of SOFTWARE LICENCE AND DEVELOPMENT AGREEMENT for definitions)

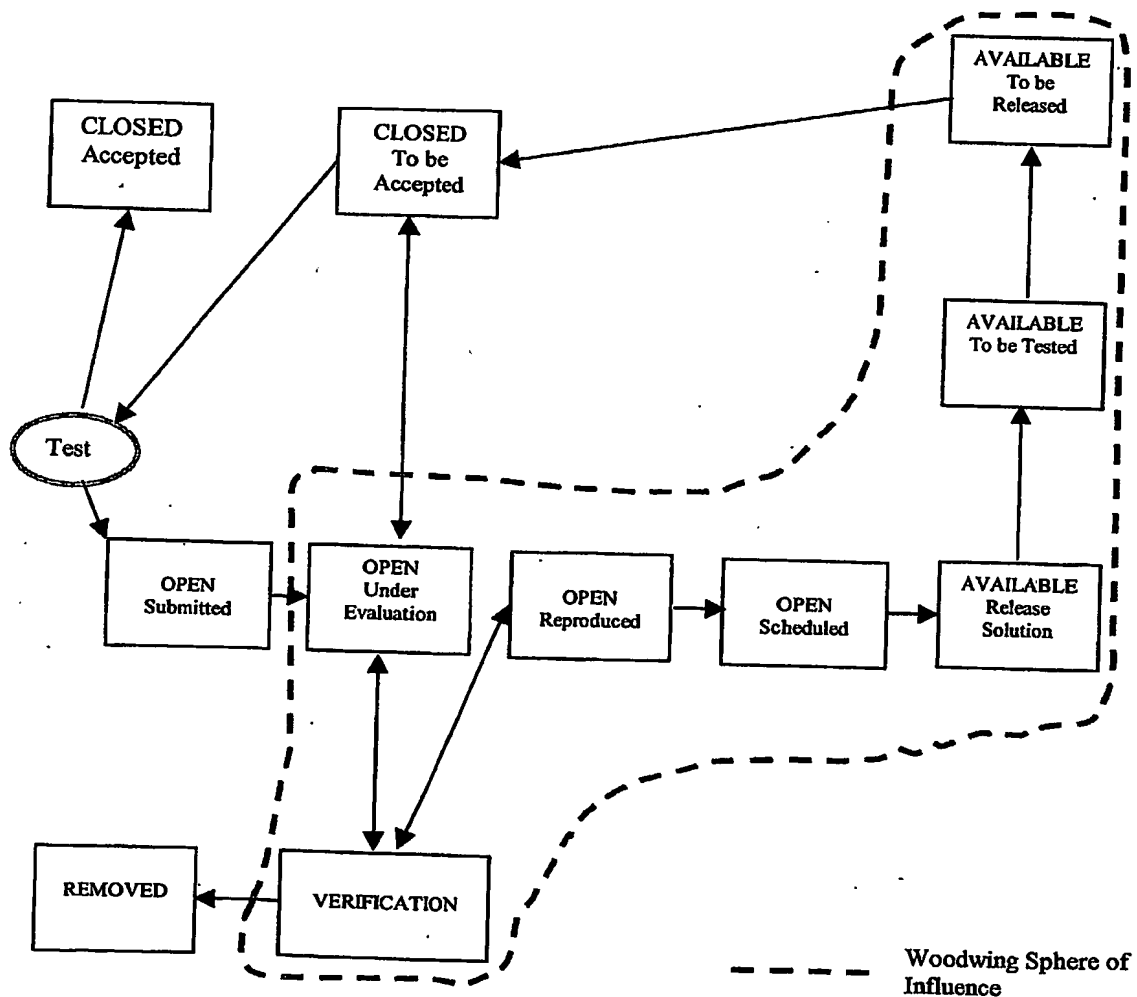
UNISYS

- Unisys shall deem that the performance of the Software is reasonable.

Fault life cycle during development phase:

Unisys will have 3 business days to install deliverables from Woodwing and respond to Woodwing outlining any faults found.

The fault life cycle is envisioned thus:





If WoodWing receives notice from Unisys of a:

Class A or B TSI, WoodWing shall issue an Update containing all fixes, or acceptable detours or workarounds if available, within 3 business days. WoodWing undertakes to commence analysis of the cause of a Class A TSI, within 1 business day of notification.

Class C TSI, WoodWing shall issue an Update containing a fix for all required errors within 3 business days for those Class C TSIs that WoodWing agrees to fix.

II. Specification/Performance Criteria of the Software

Delivery

Final delivery of all activities and work products for milestones 1 through 4 must be completed by February, 21st 2003.

Final delivery of all activities and work products for milestones 5 and 6 must be completed by March, 28th 2003.

Acceptance Test Criteria

Milestone 1

By date: 24 Dec 2002.

Contract signed

Milestone 2

By date: 17 Jan 2003. Complete Object / Page management. This includes:

- Technical design document for milestone 2
- Standard SCPro features working with Hermes:
 - Save page and object
 - Release page and object
 - Load page and object
 - Link object to page
 - Unlink object from page
 - Delete page and object
 - Multi-page handling
- Features as described in this exhibit:
3.4 - 3.6 - 3.15 - 4.4

Milestone 3

By date: 7 Feb 2003. Query panel completed. This includes

- Technical design document for milestone 3
- Standard SCPro features working with Hermes:
 - Move object and pages
- Features as described in this exhibit:
3.2 - 3.5 - 3.10 - 3.13 - 4.1 - 4.5

Milestone 4

By date: 21 Feb 2003

UNISYS

00457277.032503

- Technical design document for milestone 4
- Features as described in this exhibit:
3.1 - 3.3 - 3.9 - 3.15 - 4.3 - 4.6

Milestone 5

By date: 14 Mar 2003. Messaging and Event manager

- Technical design document for milestone 5
- Features as described in this exhibit:
3.7 - 3.8 - 3.11 - 3.12 - 4.2

Milestone 6

By date: 28 Mar 2003

- Technical design document for milestone 6
- Final integration testing
- User, install and config documentation
- Features as described in this exhibit:
4.7
- Hermes Connector Version 1 released

III. Operating System

The software developed under this contract will work properly under the following operating systems:

1. Windows 2000 Professional
2. Windows XP Professional
3. MACOS X v10.2.x

Migration to later revisions of these operating systems is discussed in Exhibit 4.



Database Format Field Design Specification

HE70-DS-DFF.doc

Authorized for issue by:

G Colagrossi
Unisys
Integration Plan
Project Manager

Signature

Date

A Politi
Unisys
Adobe Integration and
Integration Platform
Project Manager

Signature

Date

December 12, 2002
Version 1.0

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Database Format Field - Design Specification				
Author	A. Politi Integration Platform and Adobe Integration Project Manager Unisys				
To Be Approved By	AP Integration Platform and Adobe Integration Project Manager Unisys				
Comments	This is the Design Specification for the Database Format Field project				
File Name	HE70-DS-DFF.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1	All	New	12/12/2002	BC	First version.

Table Of Contents

1	SCOPE	5
1.1	DOCUMENT OVERVIEW	5
1.2	SYSTEM OVERVIEW	5
2	REFERENCED DOCUMENTS	5
3	SYSTEM DESIGN PROPOSAL	6
3.1	AFFECTED CLIENT APPLICATIONS.....	6
3.2	AFFECTED LIBRARIES	6
4	PERFORMANCE	7
4.1	SPEED	7
4.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY.....	7
4.3	CAPACITY	7



1 Scope

1.1 Document Overview

This document describes the choice of implementing the **format** field on the database.

In order to have a complete integration with third party applications the Hermes database and client applications need to be modified.

1.2 System Overview

Third party applications store their content in Hermes by placing the native format file onto the file system. Hermes clients need to show the content of the pages and objects generated by such applications.

To achieve this goal, the third party application generates a preview of the page/object and Hermes stores it along with the native file.

The originating application is always able to reload the native file to modify it. This operation can be completed by an external application via a SOAP call to retrieve the page / object.

However the Hermes Explorer application need to perform operations such as searching for pages/objects of a particular format and showing the information relevant to the originator in the query result pane.

2 Referenced documents

Official registered mime-types <http://www.iana.org/assignments/media-types/>



3 System design proposal

It could be possible to store the information relevant to the originating application in the newsroom content field of the database (pages table) but in this way, whenever a query is executed to search for a page/object produced by the XX application the Hermes Explorer will be forced to open and scan the content blob field.

For performances reason, this cannot be implemented.

The proposed solution implies adding a string field in the pages table of the Hermes database to store the mime type of the application that creates the external page/content.

The mime type will be used to uniquely identify the external application. For the user the mime type is not significant, so it will be mapped to a friendly name at run-time, according to the official mime type specification list maintained by IANA.

As a consequence, some Hermes client applications and Hermes libraries need to be modified by taking into consideration the **format field**. These changes will be made only for the internal 7.0 version.

3.1 Affected client applications

Hermes Explorer needs to be changed. In particular, it must be possible to:

- Search for a page/object by specifying the format in the search criteria
- View the format field in the page query results list.

3.2 Affected libraries

Affected libraries are

- Hrdbcore
- Hrdbcli

Both modules will allow performing the query by taking into consideration the new format field, too.

4 Performance

4.1 Speed

No degradation in speed due to the implementation of the system.

4.2 Reliability, availability and maintainability

No degradation in reliability, availability and maintainability due to the implementation of the system.

4.3 Capacity

No degradation in capacity due to the implementation of the system.

Hermes Connector Use Case Specification

Version 1.0

Revision History

Date	Version	Description	Author
8 January 2003	1.0 Draft	First draft	JH
13 January 2003		Changes after review	JH

Table of Contents

INDESIGN USE CASES	1
1. Overview	1
2. Login	2
Basic Flow of Events	2
Alternative Flows	2
Login fails	2
Preconditions	2
Post conditions	2
Sequence Diagram	2
3. Logout	2
Basic Flow of Events	3
Preconditions	3
Post conditions	3
Sequence Diagram	3
4. Save Page	3
Basic Flow of Events	3
Alternative Flows	3
Page not saved before or Save As	3
Preconditions	3
Post conditions	3
Sequence Diagram	4
5. Open Page	4
Basic Flow of Events	4
Alternative Flows	5
Page locked	5
Preconditions	5
Post conditions	5
Sequence Diagram	5
6. Release Page	5
Basic Flow of Events	5
Alternative Flows	5
Release from query result	5
Preconditions	5
Post conditions	6
Sequence Diagram	6
7. Close Page	6
Basic Flow of Events	6
Alternative Flows	6
Modified page	6
Preconditions	6
Post conditions	7
Sequence Diagram	7
8. Delete Page	7
Basic Flow of Events	7
Alternative Flows	7
Locked page	7
Preconditions	7
Post conditions	7

	Sequence Diagram	7
9.	Assign Object	8
	Basic Flow of Events	8
	Alternative Flows	8
	Preconditions	8
	Post conditions	8
	Sequence Diagram	9
10.	Check-in Object	9
	Basic Flow of Events	9
	Preconditions	9
	Post conditions	9
	Sequence Diagram	10
11.	Checkout Object	10
	Basic Flow of Events	10
	Preconditions	10
	Post conditions	10
	Sequence Diagram	11
12.	Refresh Object	11
	Basic Flow of Events	11
	Preconditions	11
	Post conditions	11
	Sequence Diagram	11
13.	Release Object	12
	Basic Flow of Events	12
	Preconditions	12
	Post conditions	12
	Sequence Diagram	12
14.	Delete Object	12
	Basic Flow of Events	12
	Alternative Flows	13
	Locked object	13
	Preconditions	13
	Post conditions	13
	Sequence Diagram	13
15.	Place Object	13
	Basic Flow of Events	13
	Alternative Flows	13
	Object already placed	13
	Preconditions	13
	Post conditions	13
	Sequence Diagram	14
16.	Detach Object	14
	Basic Flow of Events	14
	Preconditions	14
	Post conditions	14
	Sequence Diagram	14
	INCPY USE CASES	16
1.	Overview	16
2.	Login	16
	Basic Flow of Events	16

	Alternative Flows	16
	Login fails	16
	Preconditions	17
	Post conditions	17
	Sequence Diagram	17
3.	Logout	17
	Basic Flow of Events	17
	Preconditions	17
	Post conditions	17
	Sequence Diagram	18
4.	Save Object	18
	Basic Flow of Events	18
	Alternative Flows	18
	New document	18
	Preconditions	18
	Post conditions	18
	Sequence Diagram	18
5.	Open Object	19
	Basic Flow of Events	19
	Alternative Flows	19
	Locked object	19
	Preconditions	19
	Post conditions	19
	Sequence Diagram	20
6.	Close Object	20
	Basic Flow of Events	20
	Alternative Flows	20
	Opened object	20
	Preconditions	20
	Post conditions	20
	Sequence Diagram	20
7.	Release Object	21
	Basic Flow of Events	21
	Alternative Flows	21
	Opened object	21
	Locked object	21
	Preconditions	21
	Post conditions	21
	Sequence Diagram	22
8.	Delete Object	22
	Basic Flow of Events	22
	Alternative Flows	22
	Locked object	22
	Preconditions	22
	Post conditions	22
	Sequence Diagram	23

InDesign Use Cases

1. Overview

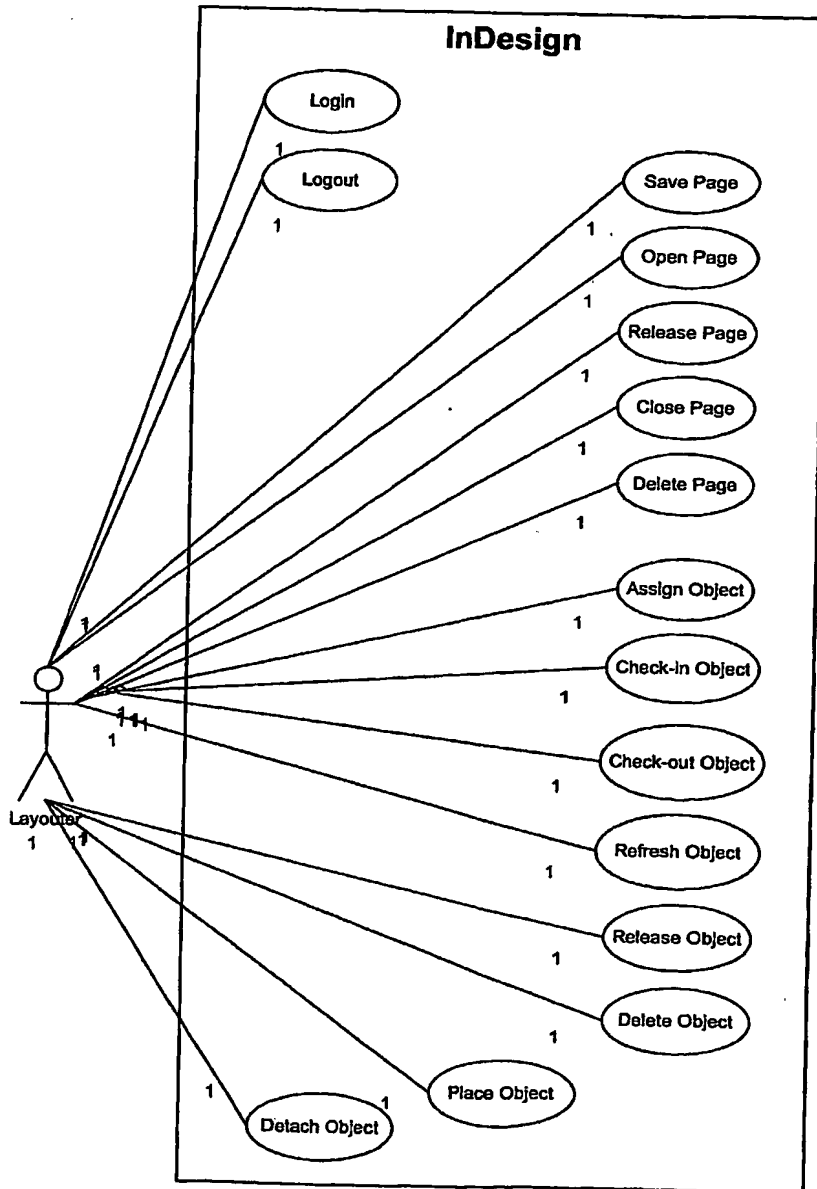


Figure 1: InDesign Use Cases

2. Login

This use case allows InDesign to log into the Hermes system.

Basic Flow of Events

1. The page designer selects the Login action
2. The system displays a login form
3. The page designer enters login information
4. The system sends a login request to the Hermes system

Alternative Flows

Login fails

If the login request fails, the page designer is informed of the failure and is given the opportunity to re-enter login information.

Preconditions

- The user is not logged in.

Post conditions

- The user is logged in; a session has been created.

Sequence Diagram

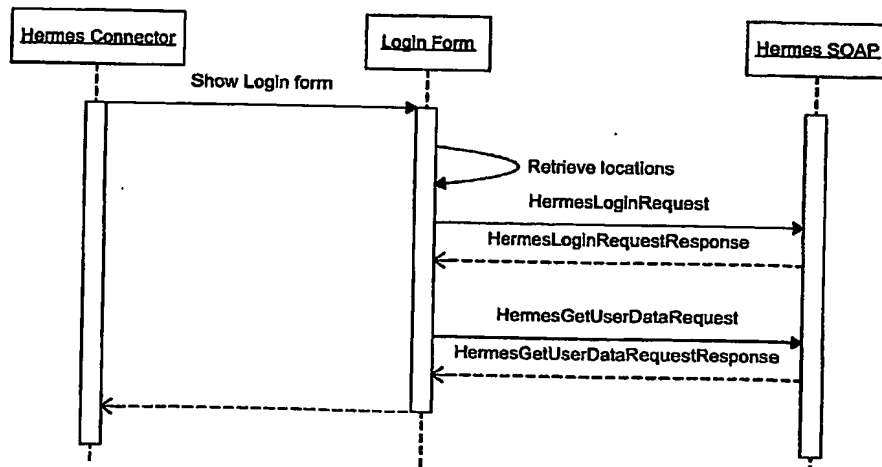


Figure 2: Login sequence

3. Logout

This use case allows InDesign to log out from the Hermes system.

**Basic Flow of Events**

1. The page designer selects the Logout action
2. The system sends a logout request to the Hermes system

Preconditions

- The user is logged in.

Post conditions

- The user has been logged out;
- Hermes actions can no longer be performed.

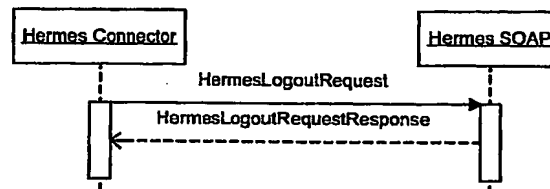
Sequence Diagram

Figure 3: Logout sequence

4. Save Page

This use case saves a Hermes page.

Basic Flow of Events

1. The page designer selects the Save action
2. The system sends a request to the Hermes system

Alternative Flows**Page not saved before or Save As**

When page has not been created using New Hermes page, or when the page designer chooses the Save As action, the system displays a similar form as with New Hermes page use case.

Preconditions

- The page designer has been logged in.
- A page is open and modified

Post conditions

- The page has been saved in the Hermes system.
- The page remains open. 

Sequence Diagram

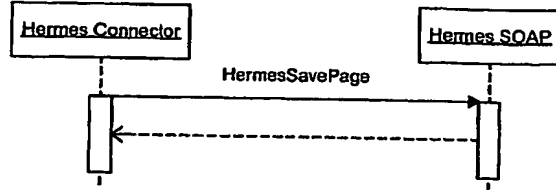


Figure 4: Save page sequence

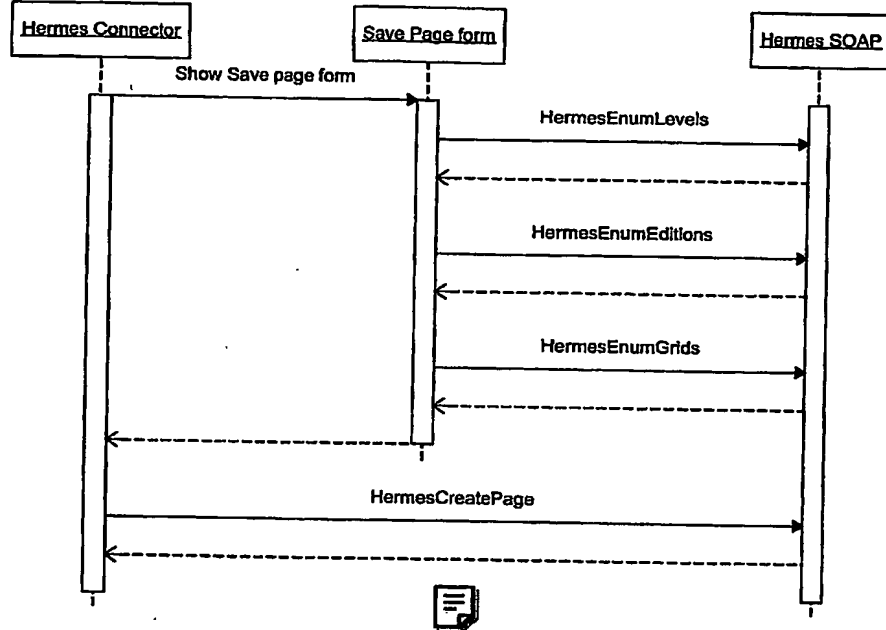


Figure 5: Save page sequence – alternative

5. Open Page

This retrieves a page from the Hermes system to edit it.

Basic Flow of Events

1. The page designer selects the Open action on a selected page in the view
2. The system sends a request to retrieve the page from the Hermes system
3. The system opens the page.

Alternative Flows

Page locked

When another user is editing the page, the page designer will be informed that the page is already in use and it will be opened read-only.

Preconditions

- The user has been logged in.
- View contains InDesign pages.

Post conditions

- The page has been opened.
- The page is locked in the Hermes system.

Sequence Diagram

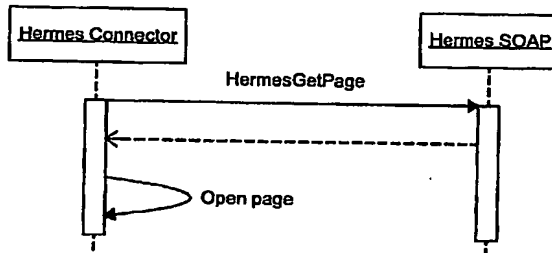


Figure 6: Open page sequence

6. Release Page

This use case changes the status of the page.

Basic Flow of Events

1. The page designer selects the Release page action
2. The system displays the Release page form
3. The page designer selects the next status from a list of available statuses
4. The system sends a request to change the status

Alternative Flows

Release from query result

When the page designer uses the Query list to release a page, the page must not be locked.



Preconditions

- The page designer has been logged in.

- A Hermes page has been opened.

Post conditions

- The status of the page has been changed.
- The document is closed.

Sequence Diagram

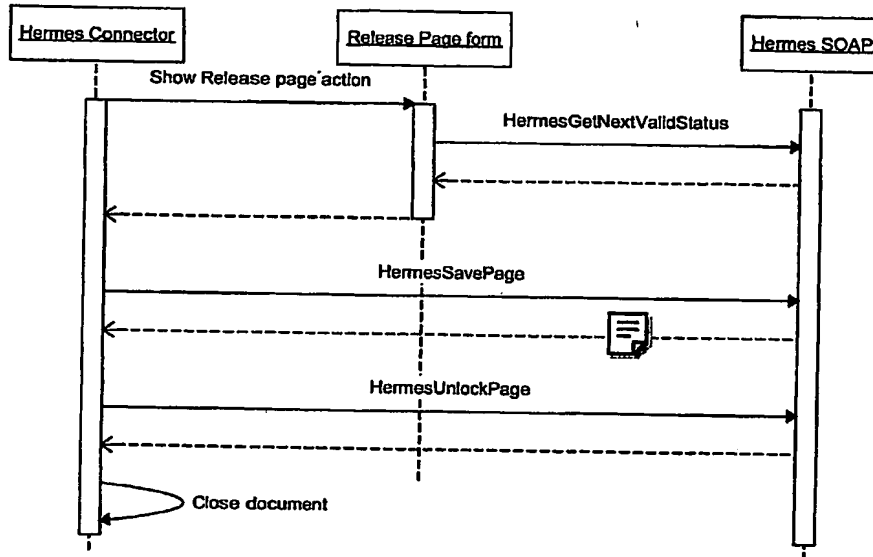


Figure 7: Release page sequence

7. Close Page

This use case closes a Hermes page.

Basic Flow of Events

1. The page designer chooses the Close page action
2. The system sends a unlock request to the Hermes system.

Alternative Flows

Modified page

When the page is modified, the page designer is asked if the changes should be saved.



Preconditions

- The page designer has been logged in.
- A page has been opened

Post conditions

- The Hermes page is unlocked.
- The page has been closed.

Sequence Diagram

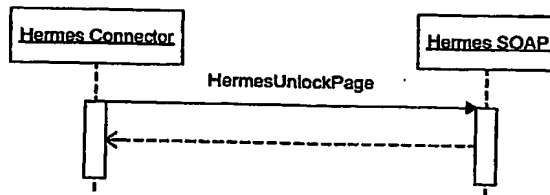


Figure 8

8. Delete Page

This use case removes a Hermes page from the system.

Basic Flow of Events


3. The page designer selects a page
4. The page designer chooses the Delete page action
5. The system sends a delete request to the Hermes system.

Alternative Flows

Locked page

When the page is locked, the page cannot be deleted.

Preconditions

- The page designer has been logged in.
- The page has not been locked. 

Post conditions

- The Hermes page has been deleted.

Sequence Diagram

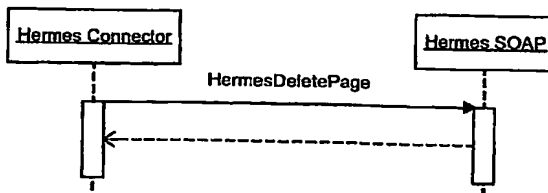


Figure 9: Delete page sequence

9. Assign Object

This use case creates a new Hermes InCopy object


Basic Flow of Events

1. The page designer selects the Assign object action.
2. The system present the Assign form
3. The page designer selects information
4. The system sends a new object request to the Hermes system.
5. The system sends a request to link the object to the page.

Alternative Flows

If an object with the specified name already exists, the system displays an error message and allows the page designer to change the name and resend the request.

Preconditions

- The page designer has been logged in.
- The page is known in the Hermes system.
- One or more stories have been selected 

Post conditions

- A new InCopy object has been created.
- The InCopy object not editable on the page

Sequence Diagram

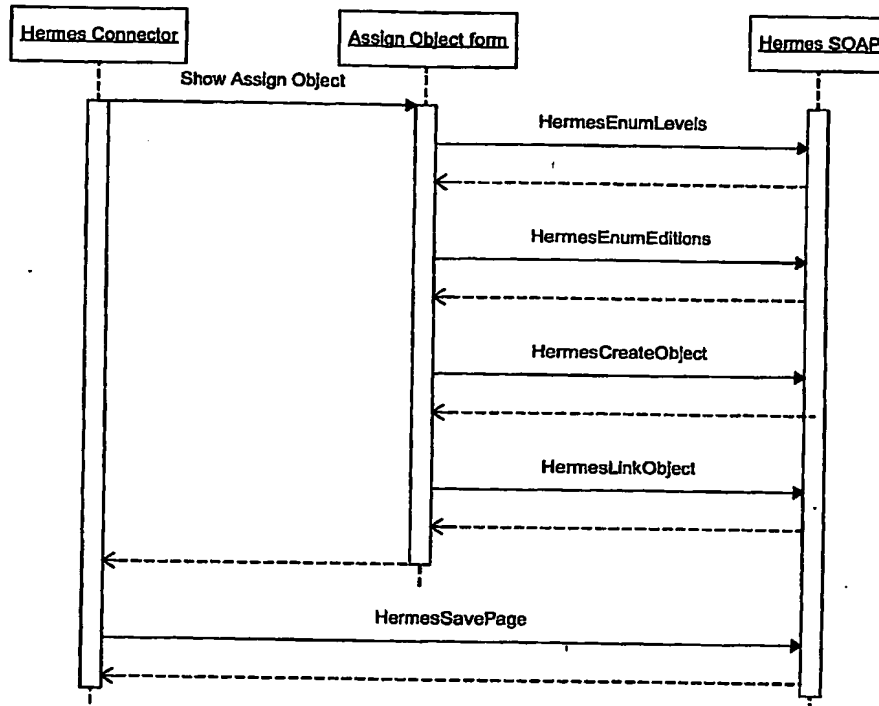


Figure 10: Assign object sequence

10. Check-in Object

This use case checks in an InCopy object linked to an InDesign page

Basic Flow of Events

1. The page designer selects the Save object action.
2. The system shows the Save object form
3. The page designer enters information
4. The system sends a request to the Hermes system.



Preconditions

- Page designer has been logged in.
- An InCopy object on a page has been selected.

Post conditions

- The InCopy object has been checked in and is no longer editable on the page.

Sequence Diagram

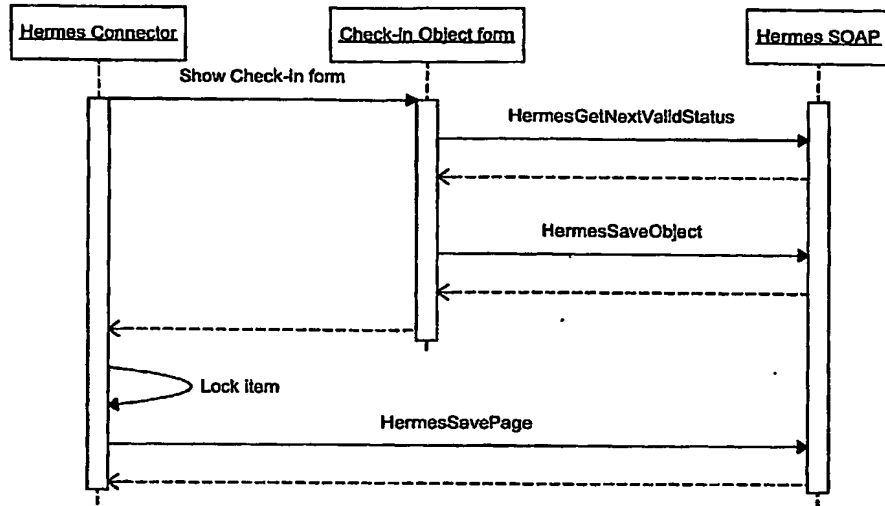


Figure 11: Save object sequence

11. Check-out Object

This use case checks out an object on an InDesign page.

Basic Flow of Events

1. The page designer chooses the Check-out object action
2. The system sends a Get request to the Hermes system.
3. The system refreshes the content of the page.
4. The system makes the Item on the page editable.

Preconditions

- The page designer has been logged in.
- An object has been selected.
- The object has not been locked.

Post conditions

- The object has been refreshed.
- The object is locked.

Sequence Diagram

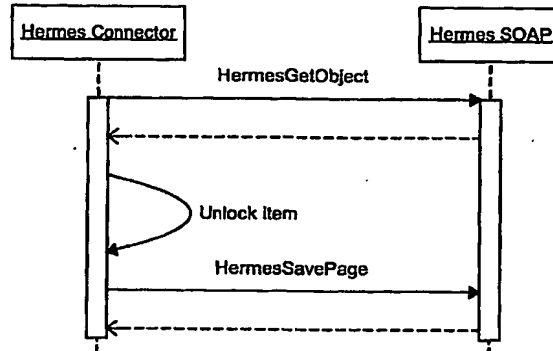



Figure 12: Check-out object sequence

12. Refresh Object

This use case refreshes an object on an InDesign page. 

Basic Flow of Events

1. The page designer chooses the Load object action
2. The system sends a Load request to the Hermes system.
3. The system refreshes the content of the page.

Preconditions

- The page designer has been logged in.
- An object has been selected.
- The object has not been locked.

Post conditions

- The object has been refreshed.
- The object is locked.

Sequence Diagram

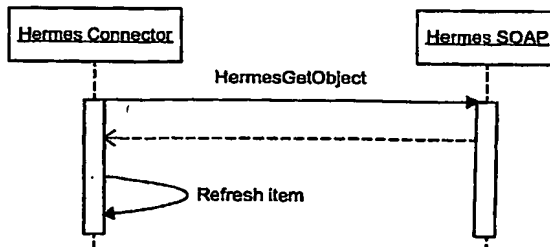


Figure 13: Refresh object sequence

13. Release Object

This use case changes the status of the object.

Basic Flow of Events

1. The page designer selects the Release object action.
2. The system shows the Release object form.
3. The page designer enters the needed information.
4. The system sends a request to the Hermes system.

Preconditions

- The page designer has been logged in.
- An object has been selected.

Post conditions

- The object's status has been changed.

Sequence Diagram

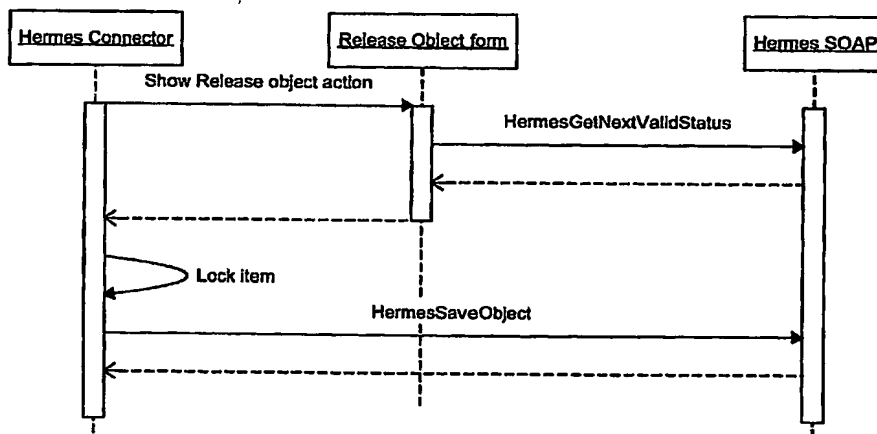


Figure 14: Release object sequence

14. Delete Object

This use case removes a Hermes object from the system.

Basic Flow of Events

1. The page designer selects an object
2. The page designer chooses the Delete object action
3. The system sends a delete request to the Hermes system.

Alternative Flows

Locked object

When the object is locked, the object cannot be deleted.

Preconditions

- The page designer has been logged in.



Post conditions

- The Hermes object has been deleted.

Sequence Diagram

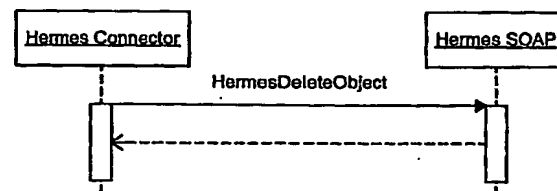


Figure 15: Delete object sequence

15. Place Object

This use case places an object onto a page

Basic Flow of Events

- The page designer selects an object.
- The page designer chooses the place action.
- The system sends a Load object request.
- The system places the object.
- The system sends a Link request.

Alternative Flows

Object already placed

If an object has already been linked to a page and cannot be linked to multiple pages, the linking fails.

Preconditions

- The page designer has been logged in
- A Hermes page has been opened.

Post conditions

- The object has been placed.
- The object has been linked to the page.

Sequence Diagram

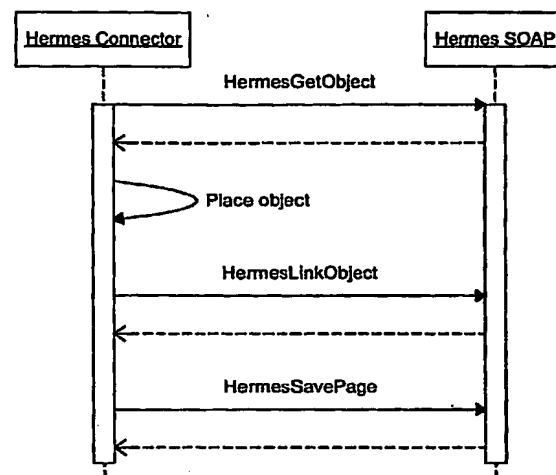


Figure 16: Place object sequence

16. Detach Object

This use case removes the link between an object and a page.

Basic Flow of Events

1. The user selects the Detach action.
2. The system sends an Unlink request.

Preconditions

- The page designer has been logged in.
- An object has been selected.

Post conditions

- The object has been detached.



Sequence Diagram

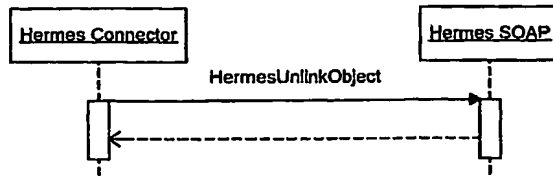


Figure 17: Detach object sequence

InCopy Use Cases

1. Overview

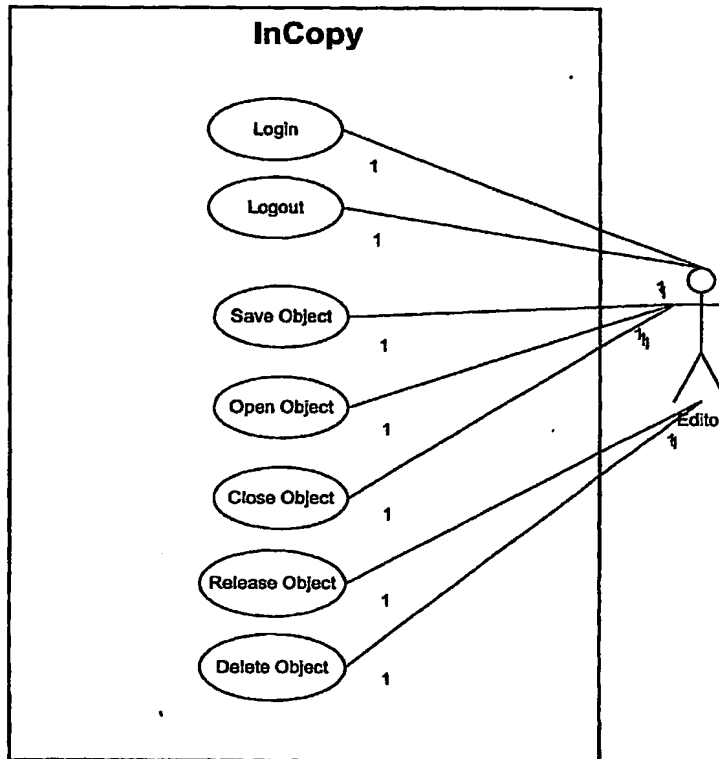


Figure 18: InCopy Use Cases

2. Login

This use case allows InCopy to log into the Hermes system.

Basic Flow of Events

1. The editor selects the Login action
2. The system displays a login form
3. The editor enters login Information
4. The system sends a login request to the Hermes system

Alternative Flows

Login fails

If the login request fails, the editor is informed of the failure and is given the opportunity to re-enter login information.

Preconditions

- The editor is not logged in.

Post conditions

- The editor is logged in; a session has been created.

Sequence Diagram

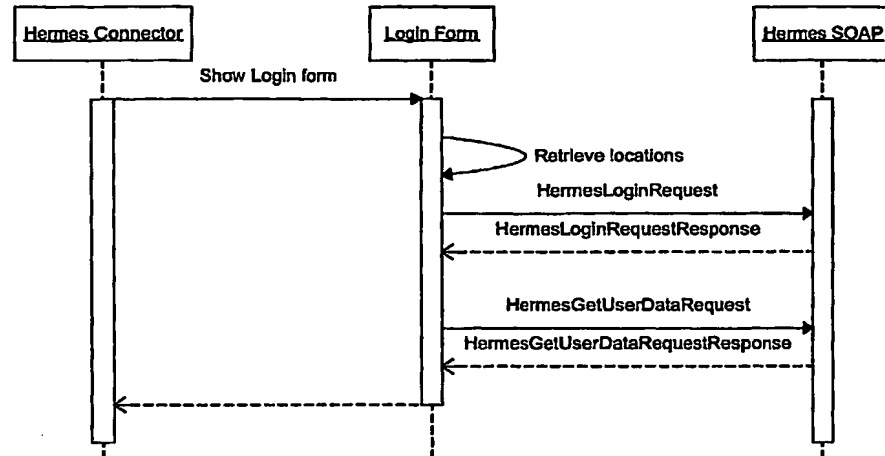


Figure 19: Login sequence

3. Logout

This use case allows the editor to log out from the Hermes system.

Basic Flow of Events

1. The editor selects the Logout action
2. The system sends a logout request to the Hermes system

Preconditions

- The editor has been logged in.

Post conditions

- The editor has been logged out; Hermes actions can no longer be performed.

Sequence Diagram

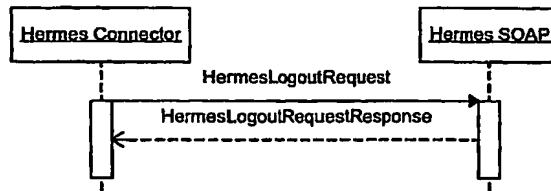


Figure 20: Logout sequence

4. Save Object

This use stores an InCopy object in the Hermes system.

Basic Flow of Events

1. The editor selects the Save object action.
2. The system sends a request to the Hermes system.

Alternative Flows

New document

If an object has not been saved before a form will be presented to the user, comparable to the New page form.



Preconditions

- Editor has been logged in.
- An InCopy object has been opened.

Post conditions

- A new version of the InCopy object has been stored in the Hermes system.

Sequence Diagram

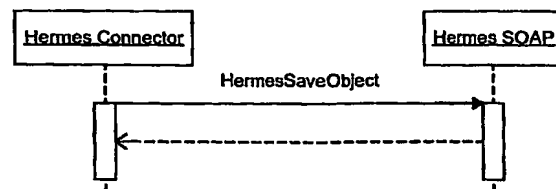


Figure 21: Save object sequence

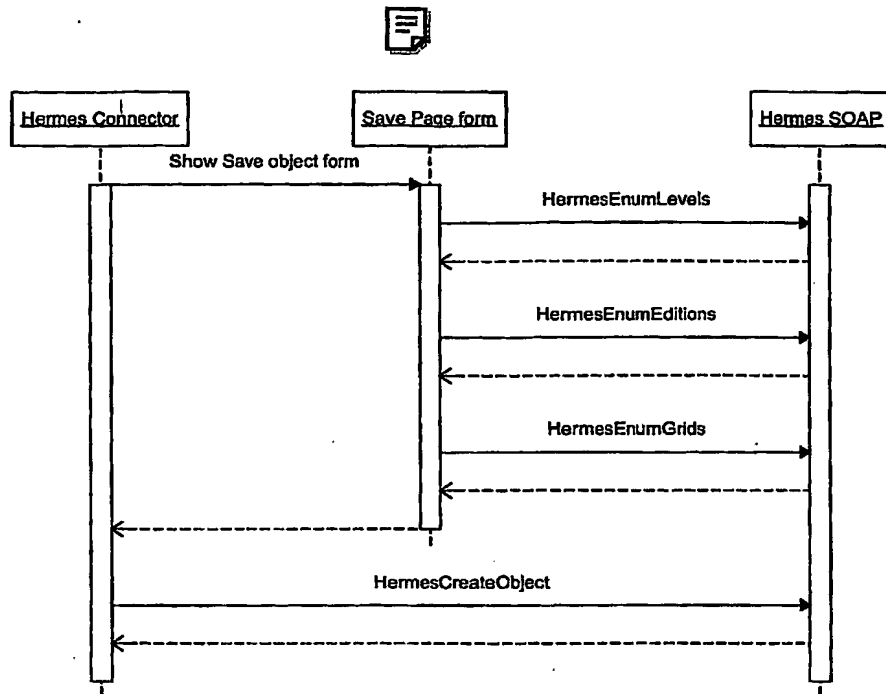


Figure 22: Save object sequence – alternative

5. Open Object

This use case opens an InCopy object.

Basic Flow of Events

1. The editor chooses the Open object action
2. The system sends a Get request to the Hermes system.
3. The system opens the object.

Alternative Flows

Locked object

If the object is locked, it will be opened read only.

Preconditions

4. The editor has been logged in.
5. An object has been selected.

Post conditions

- The object has been opened.
- The object is locked on Hermes

Sequence Diagram

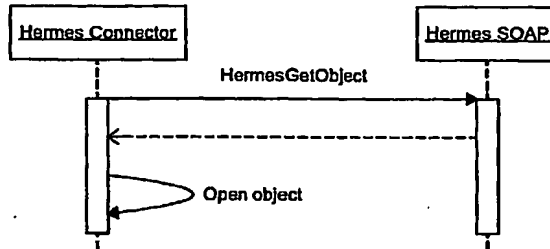



Figure 23: Open object sequence

6. Close Object


This use case changes the status of the object. 

Basic Flow of Events

1. The editor selects the Close object action.
2. The system sends a unlock request to the Hermes system.

Alternative Flows

Opened object

If the object is modified, the editor is asked if the object should be saved before closing. 

Preconditions

- The page designer has been logged in.
- An object has been opened.

Post conditions

- The object has been closed.
- The object has been unlocked in the Hermes system.

Sequence Diagram

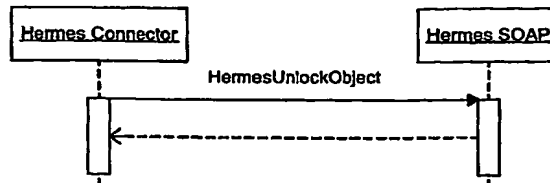


Figure 24: Close object sequence

7. Release Object

This use case changes the status of the object.

Basic Flow of Events

3. The editor selects the Release object action.
4. The system shows the Release object form.
5. The editor enters the needed information.
6. The system sends a request to the Hermes system.

Alternative Flows

Opened object

If the object is currently opened by the editor, the object will be saved and closed, before it's released.

Locked object

Objects locked by entities other than the editor, cannot be released by the editor.



Preconditions

- The page designer has been logged in.
- An object has been selected.

Post conditions

- The object's status has been changed
- If opened by the editor, the document has been saved and closed.

Sequence Diagram

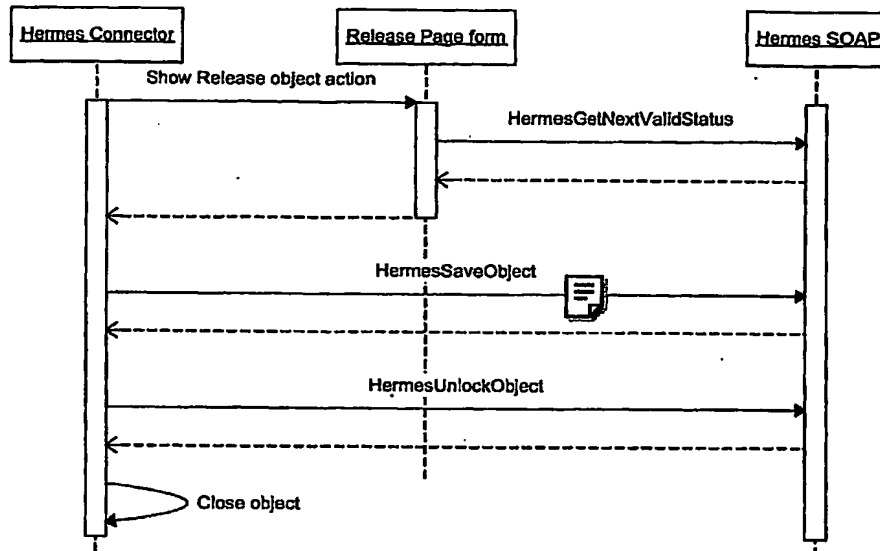


Figure 25: Release object sequence

8. Delete Object

This use case removes a Hermes object from the system.

Basic Flow of Events

1. The editor selects an object
2. The editor chooses the Delete object action
3. The system sends a delete request to the Hermes system.

Alternative Flows

Locked object

When the object is locked, the object cannot be deleted.

Preconditions

- The editor has been logged in. 

Post conditions

- The Hermes object has been deleted.

Sequence Diagram

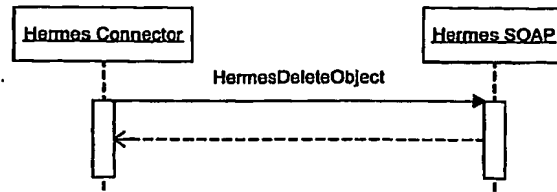


Figure 26: Delete object sequence



SOAP Application Server Project Design Specification

HE70-DS-SAS.doc

**Authorized
for issue by:**

**G Colagrossi
Unisys
Integration Plan
Project Manager
Unisys**

Signature

Date

**A. Politi
Adobe Integration
and Integration
Platform Project
Manager
Unisys**

Signature

Date

**December 12, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	SOAP Application Server Design Specification				
Author	AP Adobe Integration and Integration Platform Project Manager Unisys MB Analysis and Design Team Unisys				
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys				
Comments	This is the Design Specification for the SOAP Application Server project				
File Name	HE70-DS-SAS.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	12/12/2002	BC	First version of the Design Specification



Table Of Contents

1	INTRODUCTION	5
1.1	PURPOSE	5
1.2	SCOPE	5
1.3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	5
2	REFERENCED DOCUMENTS	6
3	CONSTRAINTS	7
4	LOGICAL VIEW	8
4.1	SUBSYSTEM DESIGN	8
5	SAS AND SOAP CORE	9
5.1	SOAP ACTIONS	9
6	EVENT DISPATCHING, MAIL AND ALERTS	12
6.1	SPECIAL CONSIDERATIONS	12
7	DEPLOYMENT VIEW	12

1 Introduction

1.1 Purpose

The purpose of this document is the definition of the architecture and design of the SOAP Application Server that empowers the current Hermes solution and minimizes the effort needed for creating a new server side component.

1.2 Scope

Scope of this document is the SOAP Integration Platform project, whose design specification is referenced later

1.3 Definitions, Acronyms and Abbreviations

ACRONYM	DEFINITION	REFERENCE
NCM-HERMES	NEWS CONTENT / MANAGER	
NGM- WIRECENTER	NEWS GATHERING MANAGER	
SAS	SOAP APPLICATION SERVER	
LS	LOGINSERVER	Security and license manager for Hermes
SOAP	Simple Object Access Protocol	http://w3.org/SOAP
XML	Extensible Markup Language	http://w3.org
HTTP	Hyper Text Transmission Protocol	http://w3.org/Protocols
HTTP-S	Hyper Text Transmission Protocol – with data encryption	http://w3.org/Protocols
SOAP	Simple Object Access Protocol	http://www.w3.org/TR/SOAP
Hermes Agent	An HTTP bridge to the Hermes system	

CMMI	Capability Maturity Model Integrated	http://www.sei.cmu.edu/cmmi/
DOM	Document Object Model	Standard for XML navigation, parsing and construction http://w3.org/XML

2 Referenced documents

Below is the table of standards referenced in this document

Table 1 Table of referenced standards

Description	Reference
SOAP Specification version 1.3	http://w3.org/soap
OpenSSL specification and downloads	http://www.openssl.org
Multipart messages	http://ietf.org/rfc.html RFC 2387
IANA official registered mime types	http://www.iana.org
The Unified Modelling Language specification version 2.0	http://www.rational.com

Additionally, readers should refer to the following documents:

- [1] SOAP Integration Platform design specification.
- [2] SOAP Event and mailing schema file, provided later during the development

3 Constraints

There are some key requirement and system constraints that have a significant bearing on the architecture. They are:

- The SOAP application server lies on top of NCM and runs on Windows and Solaris operating systems. The operating system version will be defined by the Unisys internal certification plan.
- The SAS is released under a specific license and is an optional component of the editorial system.

The SAS is the direct interface between the external applications and the SOAP integration platform. Key responsibilities can be identified into:

- **Session Management.** The SAS manages the session related to each connected user in order to deliver requests to the integration platform for the real processing. No activity can be performed by client applications without having a valid session instantiated on the SAS.
- **Hermes Configuration Management.** NCM stores important information in the Hermes.cfg file, such as workflow statuses and transitions, publication levels. This file can be very huge and delivering it directly to the client via HTTP could be expensive in terms of performances. Furthermore, reading the hermes.cfg file requires a set of proprietary API to be released and this creates a strong dependency to Hermes. Thus, an important responsibility of the SAS is the server side management of the hermes.cfg file.
- **Event Management.** The SOAP Integration Platform enables third party applications to be part of the entire production system, which is comprehensive of the event dispatching from NCM system. Events can notify client applications about changes in the database, such as page layout modifications which is a key feature in a collaborative environment.
- **Secure Connections.** The entire HTTP handling is provided with SAS and a key requirement is the implementation of secure connections to protect sensible data travelling across the network.

This key feature will be part of the SAS core module instead of making them separate SOAP services. This assumption is due to the fact that such activities are critical and cannot be removed from the system. SOAP services will perform activities more related with the editorial interaction and they can be unplugged from the SOAP layer without affecting the rest of the layer, while removing services such as "login" does not make sense.

4 Logical view

The following picture is a high level view of a possible scenario

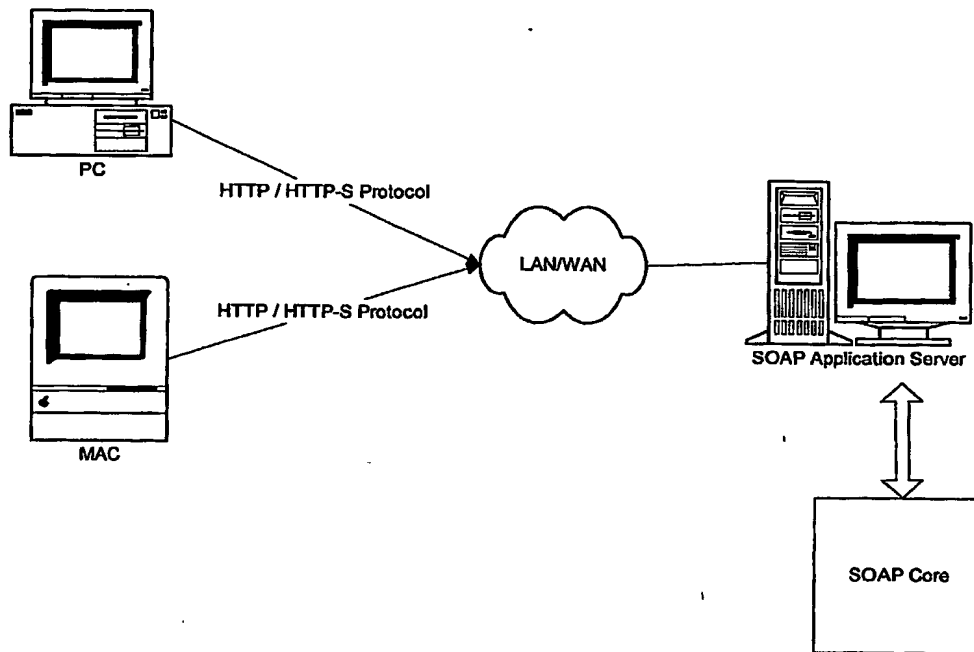


Figure 1 Overall system architecture

The new module will reuse the Hermes networking framework in order to reduce the effort and maximize the results. In fact, although different good libraries and framework are available around the world, the current Hermes Networking framework has more than ten years of test on site and, thanks to the continuous stabilization of the framework, it lets Hermes manage huge amount of data with very high-speed performances.

4.1 Subsystem design

The SAS must be a concurrent multithread server that uses a thread per connection model for processing incoming requests. The overhead required to spawn new threads is minimal, considering the performance gained for processing requests. In fact, different incoming requests and responses could have heavy load attachments that may require to be processed or sent back to



the client that could compromise the performances of other incoming/outgoing data.

Performances should be optimized using a thread pool. In fact, by using a thread pool, the overhead for spawning a new thread at each connection can be minimized or, better, concentrated at the application start-up, thus reducing latency at run-time. By using the eager spawning technique, a thread pool can be resized dynamically and provide a dynamical expansion/contraction of the pool itself, by taking into account the load that the server is managing. For example, if a thread pool is totally in use, the expansion could create a new set of N threads in the pool to fulfill the incoming requests.

5 SAS and SOAP core

The SAS works tightly close to the SOAP core implementation. Each request coming from the client application must be verified against the HTTP standard to check its compliancy. After that, the HTTP header is filtered by checking that the SOAPAction element really contains a valid value, otherwise the SAS does not allow the connection and responds with an HTTP error.

Once the request has been verified for a valid HTTP, the body of the HTTP message is sent to the SOAP core, which process the XML SOAP request. Details of the operations are available in [1].

The effort needed to build the SAS can be reduced at minimum by taking appropriate decision that leverages the current Hermes server side framework. Hermes has a very powerful networking framework that has more than ten years of field testing. Such a framework can be extended to support HTTP / HTTP-S with a small amount of coding.

5.1 SOAP Actions

SAS must support three different types of SOAPAction (see [1]). Two of them will be used for login/logout operations since the SAS directly is taking care of this critical services, while the third one will be used for all the other SOAP request that provides access to Hermes.

For any other operation, the server will route all the SOAP requests to the SOAP core after verifying that the value of SOAPAction is equal to HermesSOAP.

This apparent simplicity is the base for the evolution of the SAS and the entire SOAP Integration Platform. As an improvement, the SOAPAction could contain the URL of a different SOAPServer. The first server could act as a "macro-dispatcher" to redirect requests that can be processed somewhere else. This strategy enables a true distributed architecture with a high level of scalability and capacity.

A possible implementation of these concepts could be the extension of the SOAP platform to other applications such as NGM that runs on a different server. In this case, the SOAPAction element will contain the address of the NGM endpoint and every request will be automatically redirected to the NGM SOAP server for the processing.

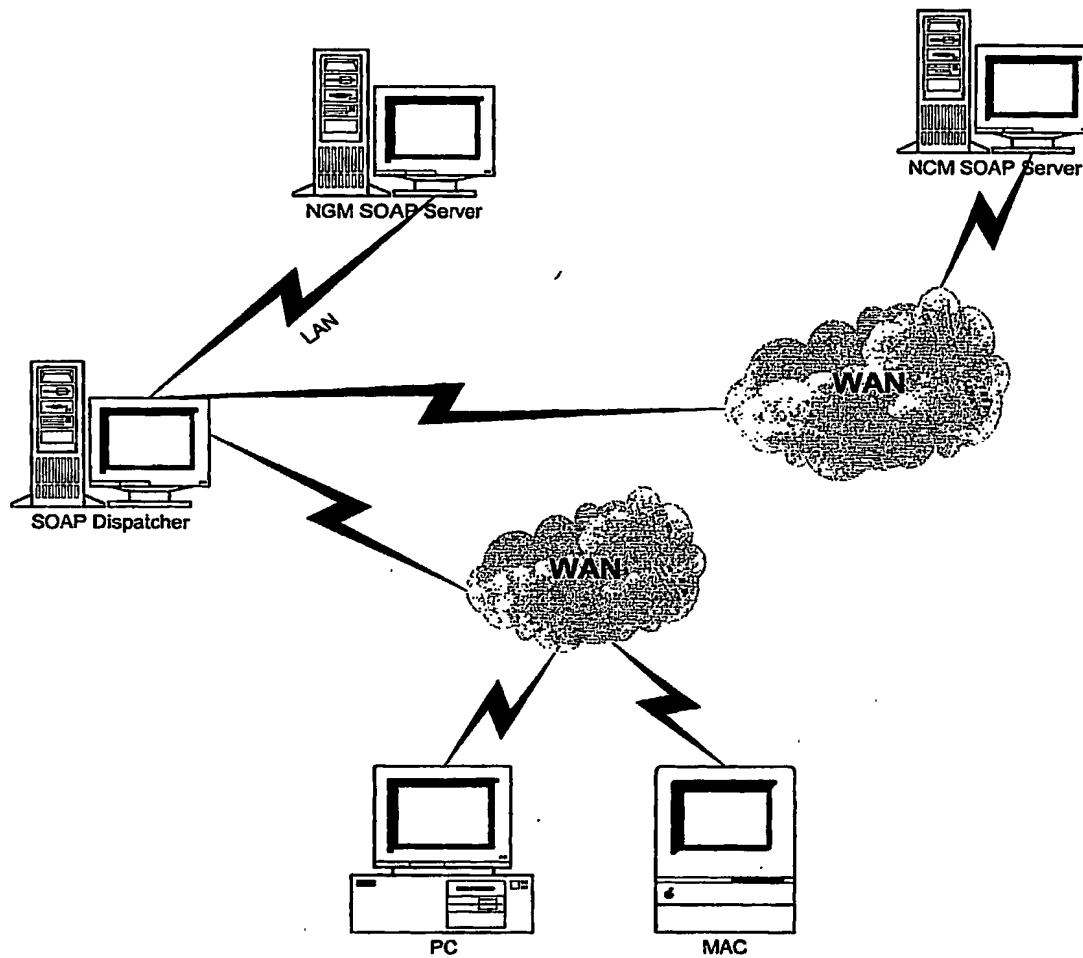


Figure 2 Sample distributed architecture built with SAS and SOAP Platform

6 Event dispatching, mail and alerts

The SAS must be able to deliver alerts, events and Hermes mail to the “client” applications. This can be easily obtained by leveraging the current Hermes Event Manager subsystem with some particular considerations. While the event management in the Hermes environment is “easy” because of the proprietary protocol and multicast availability, on the HTTP protocol this task can be trivial.

By definition, the HTTP protocol is stateless and the client application is not always connected to the server as in the Hermes LAN environment. For enabling the event, alert and mail message the concept of role-exchange can be applied. The client application will register to the SAS sending the URL and a port on which it wants to be recalled. Once an event is generated in the Hermes environment, the SAS will become the client and the registered client application will become the server. In this way, the SAS will perform a POST operation over HTTP on the URL supplied by the registered client.

6.1 Special Considerations

This mechanism must be implemented as close as possible to a “fire and forget” system. In fact, the SAS must not be blocked for waiting responses from the client application. The HTTP POST must be delivered by the SAS and it lets the server do other jobs while the client application is processing the event. If the server will wait for a response coming from the server and the client application crashes for any reason, the SAS will have a socket connection unavailable until a timeout expires, causing potential deadlock in the system and unpredictable side effects.

7 Deployment view

The installation of the SAS will be made by using the wizard normally used to install other server applications. Registry requirements will be explained when the implementation is consolidated.



SOAP Integration Platform Project Design Specification

HE70-DS-SIP.doc

**Authorized for issue
by:**

G Colagrossi

**Unisys
Integration Plan
Project Manager**

Signature

Date

A Politi

**Unisys
Integration Platform
and Adobe
Integration
Project Manager**

Signature

Date

**October 15, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

**Copyright © 2002 Unisys Corporation
All rights reserved**

Project Information

Document Title	SOAP Integration Platform Project Design Specification				
Author	A. Politi Adobe Integration and Integration Platform Project Manager Unisys				
	B. Merlo Analysis and Design Team		G.Galli Analysis and Design Team		
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys				
Comments	This is the Software Design Specification for SOAP Integration Platform				
File Name	HE70-DS-SIP.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	10/15/2002	AP	First version of the Design Specification
1.0	All	Update	19/12/2002	BM	Added SAS spec
1.0	All	Update	10/01/2003	AP	Global revision. Added collaboration diagrams

Table of Contents

1	INTRODUCTION	5
2	REFERENCES	5
3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	6
4	SYSTEM OVERVIEW	7
5	ARCHITECTURALLY SIGNIFICANT DESIGN PACKAGES	7
6	DESIGN CONSIDERATION	8
6.1	XML PARSER	8
7	ARCHITECTURAL GOALS AND DEPENDENCIES	9
8	CONSTRAINT	10
8.1	HARDWARE CONSTRAINT AND REQUIREMENTS	10
8.2	SOFTWARE CONSTRAINT AND REQUIREMENTS	10
8.3	PERFORMANCES	10
8.4	NETWORK COMMUNICATION	11
8.5	VERIFICATION AND VALIDATION	11
9	SYSTEM LOGICAL VIEW	12
10	SUBSYSTEM DESIGN.	12
10.1	SOAP APPLICATION SERVER.	12
10.2	SOAP SERVICE LAYER	12
10.2.1	Service Registry	12
10.2.2	Service entry point	14
10.2.3	Parameters Map	16
10.2.4	Attachments parameters	17
10.3	HERMES WEB SERVICES	18
10.3.1	Module caching	18
10.4	WSDL	19
10.5	SOAP RESPONSE	19
11	METADATA MANAGEMENT FOR ADOBE APPLICATIONS	20
12	EVOLUTION	20

1 Introduction

This document is the detailed design specification of the SOAP Integration Platform system. It refers to the design phase of the project 70-IPAI-SoftwareDevelopmentPlan.

2 References

The following table shows the references to the standards used during the design.

Table 1 Referenced standards

Description	Reference
SOAP Specification version 1.3	http://w3.org/soap
OpenSSL specification and downloads	http://www.openssl.org
Multipart messages	http://w3.org
IANA official registered mime types	http://www.iana.org
OPI 1.3 specification	http://partners.adobe.com/asn/developer/pdfs/tn/OPI_13.pdf
OPI 2.0 Specification	http://partners.adobe.com/asn/developer/pdfs/tn/5660.OPI_2.0.pdf
Adobe XMP technology specification	http://partners.adobe.com/
The Unified Modeling Language specification version 2.0	http://www.rational.com

Reader should also refer to the following Unisys documents

Table 2 Unisys referenced documents

Description	Reference
HE70-DS-SAS	SOAP Application Server Design Specification
HE70-FS-AQTPA	Advanced Query in Third Party Applications via SOAP Integration Platform
iPlatModel.mdl (temporary name)	SOAP Integration Platform Rose Model
HE70-MS-AINCM.doc	Vision document – Solution Management requirement for third party application
HE70-DS-AIES.doc	Adobe Integration with eEditorial Solutions Design Specification

3 Definitions, acronyms and abbreviations

Table 3 Definition, acronyms and their references

ACRONYM	DEFINITION	REFERENCE
SOAP	Simple Object Access Protocol	http://w3.org/SOAP
XML	Extensible Markup Language	http://w3.org/XML
HTTP	Hyper Text Transmission Protocol	http://w3.org/Protocols
HTTP-S	Hyper Text Transmission Protocol – with data encryption	http://w3.org/Protocols
IFRA	largest international exhibition dedicated to newspaper and media technology	http://www.ifra.com
Hermes Agent	An HTTP bridge to the Hermes system	
CMMI	Capability Maturity Model Integrated	http://www.sei.cmu.edu/cmmi/
DOM	Document Object Model	Standard for XML navigation, parsing and construction http://w3.org/XML

4 System Overview

The SOAP Integration platform is a framework built on top of the existing Hermes editorial system. The purpose of this software layer is to make a large set of Hermes API available to system integrators, in order to produce content with different software that can take advantage of the Hermes editorial production system.

The design of the system relies on standard protocol for the communication and on SOAP as the lightweight protocol to enable interaction with the outside world.

Particular interest must be reserved to extensibility and modularity, in order to minimize the impacts on other components when adding functionalities. The system will be separated into two logical components:

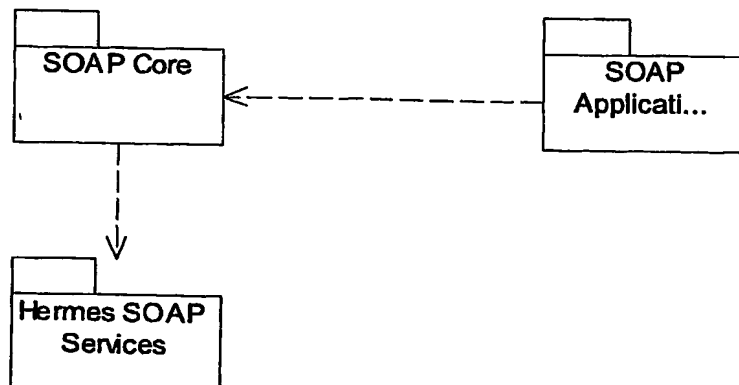
- Core Service, responsible for managing the run-time mapping from SOAP calls to WEB services and vice versa
- WEB Services, responsible of wrapping Hermes calls in a transparent way to the Core Service layer. Each request dispatched from the Core Service is rolled out by each specific WEB service.

The entire architecture includes a new Hermes module responsible of managing HTTP connections/request.

5 Architecturally Significant Design Packages

An overall component diagram of the system is shown below:

Figure 1 SOAP Integration Platform overview



6 Design Consideration

Most of the code used to write Hermes is C/C++ code and, although the core has a well-defined set of APIs, it does not allow direct interaction with applications different from the clients that are part of the system.

Furthermore, Hermes uses a proprietary communication protocol that is difficult to "externalize", because it requires that a set of specific libraries on server and on the client must be available.

This requires that the client must be equipped with a software package that prevents it from being completely independent from the implementation of the server side.

The design of this framework will focus primarily on extensibility and modularity:

- **Extensibility:** by separating the core service module from the actual implementation of wrappers around Hermes API, the extensibility is achieved, since is implicit the ability of adding new services.
- **Modularity:** in order to achieve the maximum level of extensibility of the framework, the core service layer must be separated from the actual Hermes wrapper implementation and specific services will be provided by small components plugged to the framework.

The design of a SOAP Platform that lies on top of Hermes is possible without a great effort because of the similarity of the network protocol used by Hermes with the HTTP concept. In fact, although they are different in constitution, both rely on the request response strategy to implement communication between client and server.

6.1 XML Parser

XML is managed through the use of the Xerces library, which offers a C++ implementation of the DOM conform to the W3C specification.

7 Architectural Goals and Dependencies

The SOAP Integration Platform will be designed to exploit the existing Hermes implementation, allowing the Hermes system to interface with any other external application. The WEB services will implement services by calling and arranging the existing Hermes APIs. Specific assumption and dependencies are:

- There will be no new development inside Hermes to adjust gaps between SOAP and the core. Instead, the design assumes as a fixed point the existence of Hermes.
- A new module will be implemented as the endpoint server for two reasons:
 - The existing hermesagent cannot be overloaded since it is used for the HermesWeb product and no modification to the performances is possible.
 - The Integration Platform is a product that Unisys will deliver under license and not as an add-on to an existing licensed product.
- No direct database operation will be performed. Each access to the Hermes storage will rely on the existing Hermes APIs, in order to leverage the current capabilities.

8 Constraint

As mentioned before, Hermes is a proprietary system built with C/C++ language. Java should be the natural way of building a SOAP extensible layer but the gap between the two languages is difficult to be managed and, when managed, it relies on a difficult to understand mechanism that cannot be easily reproduced.

8.1 Hardware constraint and requirements

The SOAP integration platform will be available for both Intel and SPARC processor.

8.2 Software constraint and requirements

For the software environment:

- Windows 2000 Advanced Server
- SUN Solaris 5.8, 32 bit version

Both platforms will be available according to the Unisys certified standards. The implementation of the code must take into consideration of the architectural differences among platforms.

8.3 Performances

The design of the framework will take into high consideration the performance issue and no degradation or limitation to the existing Hermes system will be caused by the framework itself.



8.4 Network Communication

Network communication between the Integration Platform and the various actors will rely on standard HTTP / HTTPS transport over TCP/IP protocol. The communication mechanism used inside Hermes is request-response based. Client applications post a request to the server and wait for the response from the server which does all the processing requirement. The same concept is applied to the HTTP, with the main difference in the "purpose" of the protocol. The Hermes proprietary is tailored for the specific application needs, while the HTTP is a multi-purpose generic protocol.

This means that performances are quite different, since the HTTP needs a lot of extra meta-information to describe the data it is carrying over the network. Due to this verbosity, a particular attention is paid in the optimization of long responses from the SOAP server.

The SOAP integration platform provides communications in both plain HTTP and over Secure Socket Layer by using the OpenSSL library. The secure framework implementation is, however, responsibility of the SOAP application server, which design is detailed in document HE70-DS-SAS.

8.5 Verification and validation

The entire platform will be tested with specific procedures to ensure that:

- Multiple concurrent SOAP messages can be delivered to the services
- Buffer overflow protection is correctly implemented
- Invalid XML packets are correctly handled without problems to the SOAP core
- Malformed HTTP requests are handled correctly without problems to the SOAP core

Each developed service will be placed under regression test to ensure that no alterations to the SOAP core service layer have been made.

The alpha tests will be made by using scripts to send XML calls to the framework. The scripts will be refined on use and delivered to the testing group as an add-on to help integrity test.



9 System Logical View

In order to depict a logical view for the system, a typical usage scenario is shown below.

10 Subsystem design.

10.1 SOAP Application Server.

The SOAP application server is described in document HE70-DS-SAS

10.2 SOAP Service layer

The SOAP Service Framework is responsible of the management of the SOAP requests received by the SAS. The SAS checks the HTTP header for validity of session id and SOAPAction values. If these values cannot be verified, the request is not handled and an error is returned to the caller in the form of a SOAP fault.

If the HTTP header is valid, the SAS calls the SOAPService Registry and passes the SOAP request. The SOAP service registry looks for the service handler (WEB Service) that implements the method requested via SOAP and, if a service is found, it dispatches the SOAP call to the specific WEB service for being processed. The SOAP core will offer a set of methods to the WEB services, thus allowing to:

- Get parameters from the SOAP call
- Get the value of the parameters appropriately converted to C/C++ type
- Create the SOAP response to the call and abstract any specific XML construction and handling

10.2.1 Service Registry

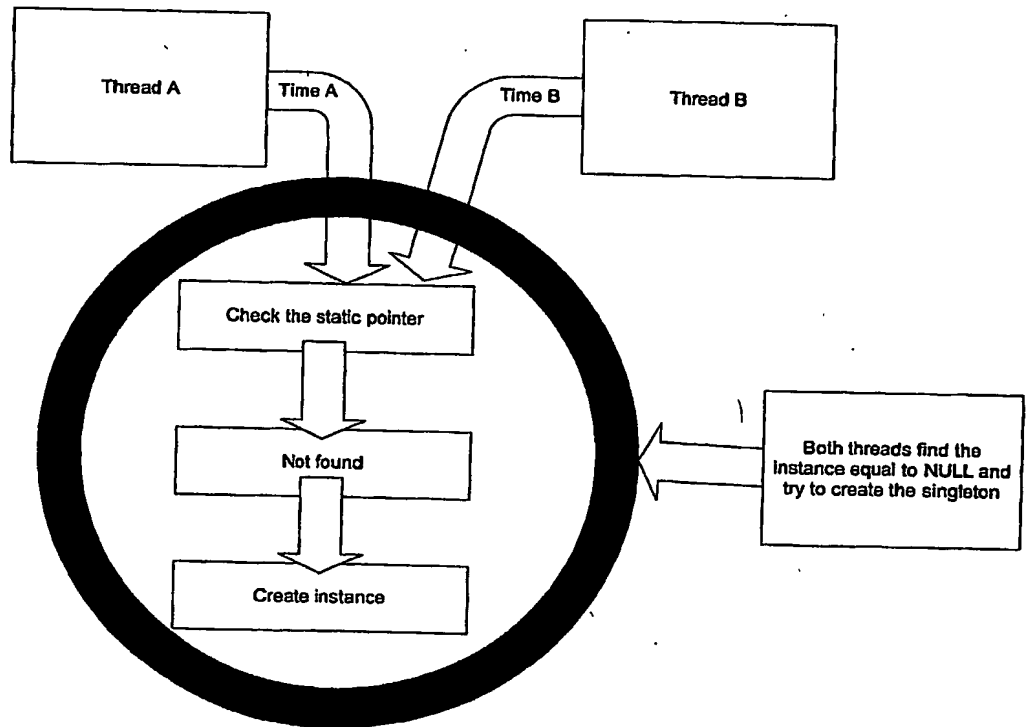
The service registry lies in the SOAP core and is responsible of maintaining the list of service and the mapping between SOAP methods and services that implement them. In order to keep a high level of extensibility, the services are mapped to specific calls via an XML file (deployment file) that is read during the server start up.

Doing so, enabling or disabling specific services will simply imply removing the mapping inside the deployment descriptor.

Furthermore, a particular protection level is placed into the service registry to allow continuing to offer services also if one or more services are deleted from the file system. The service registry is implemented as a singleton object using the Singleton pattern, in order to give the caller a single point of access to the WEB Service instantiation point.

The service registry singleton should be implemented with a double-checked locking pattern (Schmidt, 1 2001 145). Although the service core itself does not dispatch services into different threads because it relies on the multithread SAS engine, this rule could become invalid for performance reasons. By definition, the singleton exists in a single instance. The constructor of this type of object, however, cannot have a locking mechanism because of the following race condition:

Figure 2- Multithread unsafe singleton instantiation



Thread A

Thread B

Check the static pointer

No static pointer is available Check the pointer

Lock

No pointer available

When the thread A is in the LOCK condition is about to create the static pointer. When the thread B comes into, no instance is available and proceeds to its creation. To prevent this, the double-checked locking pattern [2] tests for the instance of the object and, if it does not exist, the object is locked. After the lock is acquired, the thread tests again the instance to see if in the meantime another thread created it. The maximum level of safety is achieved because the second thread waits until the mutex is available and, when this condition is true, the thread finds the instance already created and returns the pointer to the instance itself.

<< Concurrent access to the singleton – Drawing Placeholder>>

Figure 3 Thread safe singleton instantiation

After the Service registry has determined the method that is being called by parsing the SOAP request, it returns a pointer to the WEB service entry point in order to dispatch the work.

By using the double-checked locking pattern, the atomicity of the change to a singleton object is guaranteed even in multi CPU environment. In a single CPU environment, in fact, threads are “interleaved” and concurrency occurs less frequently than in multithreaded applications running on different CPUs.

The service registry is as an object factory and should be implemented as in [4] to decrease the interdependency between implementation of the core and implementation of the services and to speed up the SOAP dispatching.

10.2.2 Service entry point

In order to decrease the dependency between the core and the modules, the SOAP integration platform uses a deployment descriptor to map the service implementation to a service entry point. Whenever a request comes into, the service registry looks up the SOAP method into the XML payload and searches for a suitable implementation. If an implementation is found, its entry point is called to instantiate the actual service implementation.

C++ language does not allow a class to be created using its name as parameter, for example, to the constructor. A code such as

```
const char *className = "SpecificClass";  
  
BasePointer *instanceOfTheClass = new className;
```

Is not valid.

The solution of such a problem, which enables the ability of mapping service to method in an external file, is to use the inheritance to make all the service implementation specialization of a base class. Each implementation method inside the WEB service, has a C entry point whose name is actually mapped to the service via the deployment. The service registry reads this entry point from the deployment descriptor, loads the WEB service dynamically and gets the address of the creator function (entry point). Once this has been done, the service registry calls the entry point of the WEB service that does nothing but creating a new "class" and returning a pointer to it.

Implementation class

```
class SOAPImplementation : public SOAPServiceHandler
```

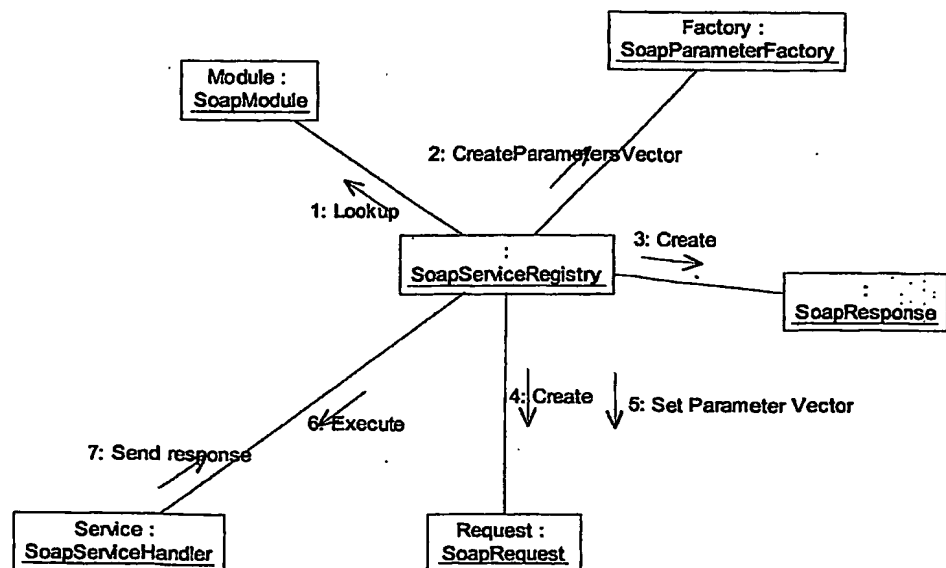
The entry point will be

```
SOAPServiceHandler *EntryPointFunction()
{
    return new SOAPImplementation;
}
```

This is the only way to bypass the previously mentioned C++ limitation. By using this technique, each SOAP call can be added to the entire framework without recompilation of the layer.

The collaboration diagram for the service lookup and dispatching is reported in Figure 4

Figure 4 Service instantiation and dispatching





The service registry maintains a mapping between the service name exported and the module that implements the function. This mapping is loaded at the service registry instantiation by reading the deployment descriptor from the file system. The deployment descriptor is an XML file that maps the name of the entry point inside the service and the dynamic library with the implementation. Since this mapping does not change during the service registry run time, it should be optimized for fast searches rather than for fast insertion. A suggestion for obtaining fast searches is to use the standard library vector instead of a map. As discussed in [3] using a sorted vector of elements is significantly faster than using a map of keyed value, since the map is implemented as a RB-tree with overhead that is optimized for fast insertion.

Once the service has been found and instantiated, the SOAP core finishes to parse the XML and loads the parameter map.

10.2.3 Parameters Map

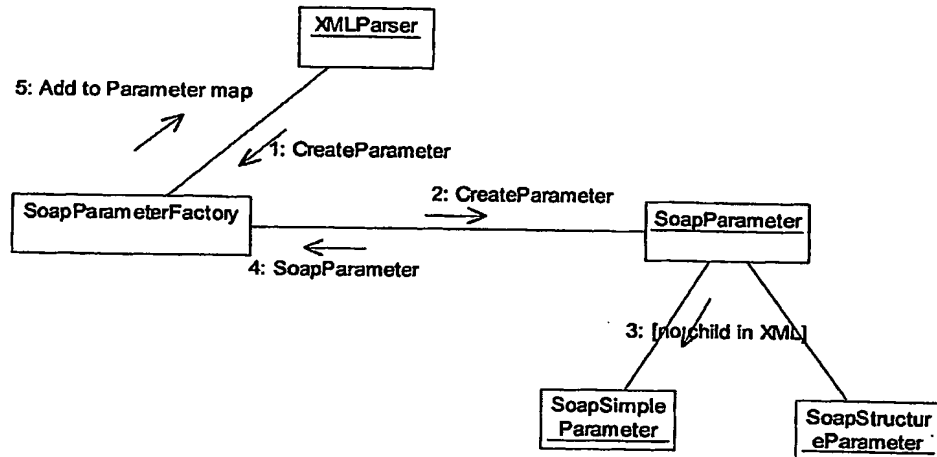
The SOAP core manages two types of parameters:

- Simple parameters
- Complex parameters (structure)

Simple parameters are used to map directly name-value pair, while complex parameters can contain simple parameters. This allows the SOAP framework to manage a large variety of parameters combination. In fact, by using a Composite Citations

[1] pattern, it is possible to manage parameter that can be composed of a structure and each element of the structure is composed by other structures. To create the correct parameter object, which depends on the XML structure of the SOAP request, a class factory Citations

[1] is used to delegate the construction of the appropriate object. The collaboration among object is represented by the following diagram:

Figure 5 Collaboration diagram for parameter creation

The parameter map should be implemented for fast retrieval rather than for fast insertion, thus the suggestion becomes the same as per the service mapping discussed in chapter 10.2.1

10.2.4 Attachments parameters

The integration platform allows creation of content with third party applications and such a content is "converted" to Hermes objects and pages. SOAP incoming messages must have the ability to attach files. To allow the mixed XML and binary content in a SOAP message, a multipart – mime message is used. The SOAP core is responsible for the parsing of the multipart message and the storage of the raw bytes of the attachments. Details of messages with attachments are provided in the HE70-DS-AIES document, where the interface between the system and client applications is fully described.



10.3 Hermes Web Services

Hermes Web Services are components that are plugged into the SOAP Integration Platform to perform the real request processing. These modules are implemented as dynamic link libraries that are loaded on demand and stay active during all the run time.

Hermes Web Services have the responsibility to wrap Hermes legacy code into object-oriented components that are instantiated by the platform. The effect on the Hermes code is null and the implementation can leverage the existing API, thus minimizing the effort to make Hermes an open system.

The services should be divided into "categories" depending on the task they run. A special category is that including services responsible of critical task, where this tasks cannot be unplugged from the platform since they are vital for the existence of the platform. In this category, the following services will be placed:

- Login service
- Logout service
- Session management service

These types of services cannot be mapped into the deployment descriptor, thus they cannot be unplugged from the platform.

Hermes specific services are divided into the following categories:

Service category	Hermes specific task
Objects	Hermes object management, such as creation, deletion, retrieval of objects.
Pages	Hermes page management, such as creation, deletion, retrieval of pages, link/unlink of objects to/from pages .
Edition	Edition management, such as edition listing, zone management
Workflow	Object and page status management
Users	User management, such as user data retrieval, permission checking
Level	Level management, such as level browsing

10.3.1 Module caching

The SOAP core manages the various WEB services by keeping a table of method/service implementation. Each time a method is run, the table is updated by incrementing the usage count for the module and decrementing each module that has not been hit. When the counter for a module is equal to 0 (zero), the module is discarded thus gaining memory and, consequently, performances.



10.4 WSDL

Each implementation module should be able to describe itself and the parameters name/types, as well as the return values. This enables the SOAP core to implement a system service that collects the information about the services. By doing this, a discovery tool can be created to give a "generic client" a way to obtain information about what Hermes SOAP is offering in term of WEB services.

10.5 SOAP response

The responsibility for creating response is equally divided between the service handler (implementation of service) and service dispatcher. The service dispatcher offers a response object to the called implementation that, in turn, fills in the return value elements that compose the response. The service dispatcher creates an XML SOAP response by attaching the response built by the service implementation to the SOAP envelope. The service dispatcher is responsible to create the correct XML response and to verify that it is conformant to the SOAP specification.

11 Metadata management for Adobe Applications

Metadata will be handled whenever possible using the XMP technology. XMP is an Adobe proprietary technology platform for metadata embedding inside data file. The XMP usage has different advantages, such as:

- Free of charge and OpenSource: this allow the developer to work with the source code which is extremely important to understand the working internals and for the debugging of application
- Cross platform. The XMP SDK is delivered and tested on different platforms, such as UNIX and Windows. This enables the complete reuse of the core code
- Metadata are embedded into asset files using XML, Dublin Core specification.

Metadata can be embedded in InDesign/InCopy file directly from the application and extracted on the server, thus allowing the storage in a database that offers query functionalities.

XMP must not be used to embed metadata in image file. In fact, the JPEG specification for example, implies that a reader application or image manipulation software discards every non JPEG data inside the file. Opening a JPEG with XMP embedded metadata with software different from Adobe will result in removing the metadata.

12 Evolution

The architecture of the SOAP integration platform and the implementation based on legacy code wrapper, enables a high level of extensibility. After a first major release of the integration platform, new services will be added to perform even News Gathering Manager operations.

From the platform perspective, an interface for service management will be created, allowing the service startup and shutdown from remote client applications, such as an administration WEB page.

This design and architecture team will consider also a model for inter-service communication, to allow the reuse of code and implementation among services.

Citations

- [1] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley 1995
- [2] Douglas Schmidt, *C++ Network programming Volume 1 and 2*, Addison-Wesley 2001
- [3] Scott Meyers, *Effective STL*, Addison-Wesley 2001
- [4] Andrei Alexandrescu, *Modern C++ Design*, Addison-Wesley 2001



Third Parties Integration with NewsRoom Design Specification

HE70-DS-TPIN.doc

Authorized for issue by:

**G Colagrossi
Unisys
Integration Plan
Project Manager**

Signature

Date

**A Politi
Unisys
Adobe Integration and
Integration Platform
Project Manager**

Signature

Date

**January 02, 2003
Version 1.0**

A large black rectangular redaction bar is located at the bottom of the page, spanning most of the width.

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Copyright © 2000 Unisys Corporation
All rights reserved

Project Information

Document Title	Third Parties Integration with NewsRoom - Design Specification				
Author	A. Politi Integration Platform and Adobe Integration Project Manager Unisys				
To Be Approved By	AP Integration Platform and Adobe Integration Project Manager Unisys				
Comments	This is the Design Specification for the Third Parties Integration with NewsRoom project				
File Name	HE70-DS-TPIN.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1	All	New	01/02/2003	BC	First version

Table Of Contents

1	SCOPE	5
1.1	SYSTEM OVERVIEW	5
1.2	DOCUMENT OVERVIEW	5
2	REFERENCED DOCUMENTS	5
3	SYSTEM DESIGN DECISIONS	6
3.1	HIGH LEVEL SYSTEM BEHAVIOR	6
3.1.1	<i>Description of pages handling</i>	6
3.2	INCOPY SPECIFIC CHANGES IN NEWSROOM	8
3.3	LIMITATIONS	9
4	PERFORMANCE	10
4.1	SPEED	10
4.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY	10
4.3	CAPACITY	10



1 Scope

1.1 System Overview

The purpose of this project is the integration of third parties applications with the Hermes editorial system. Third party applications store their content in Hermes by placing the native format file onto the file system. Hermes clients need to show the content of the pages and objects generated by such applications.

External logical pages will allow the end user to use external software such as Adobe InDesign to produce pages that will be managed by Hermes.

The intent is to define a protocol that allows current implementations to handle such pages in a transparent way, as well as the standard Hermes pages.

To achieve this goal, the Newsroom content must be modified to store the path to the EPS preview of the page.

1.2 Document Overview

This document describes how Hermes will manage the logical pages created with third party applications using the SOAP integration platform.

2 Referenced documents

HE70-FS-TPIN.DOC	Third Parties Integration with NewsRoom
HE70-DS-AIES.doc	Adobe Integration with eEditorial Solutions Project

3 System design decisions

In designing this feature, we will consider the current system architecture and empower it, in order to keep the code changes as slight as possible and leverage the current implementation of NewsRoom, UPS-Explorer and all the applications that handle logical pages and objects.

3.1 High level system behavior

The logical pages created with external applications will be stored as files in the Hermes file system in order to allow the native application to reopen it. External applications will produce a preview of the page, either in EPS or in JPEG format, that will be used by Hermes client to show the preview of the page and, generally, to overcome the impossibility to read native format files. Currently, there is no support of previews in PDF but we'll keep the focus on it since it could be a major improvement in order to achieve the complete PDF/X1A – PDF/X3 workflow adopted by major magazine producers. Hermes needs to keep track of the page in the database in order to:

- Enable the workflow for the asset
- Enable the imposition software to directly paginate it
- Enable the tracking of objects linked to external pages, allowing the current UPS-Explorer to perform queries on object linked to a specific page and vice versa

3.1.1 Description of pages handling

External application creates logical pages that will be recognized as normal Hermes logical pages. This will lead to slight modifications of the Hermes clients in order to allow the Hermes applications to handle these pages.

To show the external pages, a new set of markers will be placed into the content field to identify the data relevant to the external page. In this block of data, the same information used to archive images (called DBBUFFER) will be placed, along with the path to the EPS for the preview of the page. This allows Newsroom to transparently handle this content, even if the real page has been designed using an external application.

When Newsroom reads the content and encounters these markers, it will read the path to the EPS, create a dummy layout whose size will be exactly as the size of the EPS and perform a "link on the fly" to the EPS.

By using the distributed grid functionality of the SOAP server, an external client can produce pages having the same size as the grid, thus keeping all the dimensions consistent. This means that the choice of a particular grid will

produce an EPS exactly sized as the grid and, since grids are taken directly from the server, all the dimensions will be consistent.

All the modifications to the external page content will be possible only using the originating application, while the preview attached to the dummy layout will be placed as background of the Newsroom page and changes will be inhibited. The Newsroom page will have a layered structure, so that it is possible to add Newsroom layouts to the external page, since the content will be the same. All the new elements added to the page are standard Newsroom elements that will be visible only in NewsRoom. If the external page is reopened with the originating application, the external software will manage only the part relevant to the application itself. Since the external page is treated as a whole, Newsroom elements cannot be merged with external page elements because NewsRoom will not know the position and geometry of the elements inside the external page. This is a limit in the multiple application page composition. Clipping between elements will be possible manually. This means that, for example, if a Newsroom layout is placed in a certain position with respect to an object in the external page and this object changes its position inside the page, the adjustment of the NewsRoom layout must be corrected.

To minimize the side effects on other Hermes client applications, all the storage and management must follow the currently in place rules. An external logical page will be managed at low level in order to be viewed by NewsRoom as a normal page.

In order to identify an external page to prevent modifications, for example in layouts, the external logical page is linked to an external layout. The external layout is skipped in all the modification functions in Newsroom, thus resulting in a protected layout. However, the layout will be fully compliant with the rest of the layout types, except that it's marked as external. This homogeneity allows the tracking of the objects linked to a page, even if the object is not an internal Hermes object and the page is not an internal Hermes page. Whenever an external application links an external object to the page (for example InDesign page links an InCopy object), a notification via the Integration Platform is made to keep track of the operation.

Once the notification is received by the Integration Platform, a new "Hermes" link between the page and the object is created inside the database in the same way it's recorded when an object is linked to a page in Newsroom.

As a counterpart, when an object is unlinked from a page, a notification is made to the Integration Platform which will remove the association from the database. By using the common link/unlink protocol used by Hermes and treating pages and layouts for external application in the same manner Hermes does, it will be possible to query a page for all the linked object also for external pages.

Since unlinking objects from pages in most cases results in the removal of the object from the page, the unlink operation will be available only in the originating application.

In other words, the unlink of a photo or an InCopy text attached to and InDesign page is possible only from InDesign, considering also that the

InDesign file must be updated to remove the internal link and the knowledge of this process is available only on the client application.

However, if a server side InDesign manipulation tool will be released by Adobe in the future, the implementation of the link and unlink between objects and pages will require the specific "tool-side" implementation, without changing the mechanism currently in place on Hermes to handle external asset link.

3.2 InCopy specific changes in NewsRoom

Currently, InCopy is not able to generate a preview of the text. Furthermore, if an InCopy object is linked to an InDesign page, InDesign holds the text representation of the object. To let Hermes clients to "see" a preview of an InCopy object, a dummy preview is used.

Since the object linked to the external page appears as a link for NewsRoom, a specific adjustment must be made in order to let NewsRoom skip the visualization and printing of the dummy preview. The real content of the hidden object is already on the InDesign page.

If the InCopy object is linked to a NewsRoom native page, the content of the object is converted into NewsRoom textual object and the process can proceed as a normal object linking in NewsRoom and the dummy preview problem is automatically solved.

For each other type of external object, there is no special requirement since they will be treated by NewsRoom as image objects.

3.3 Limitations

All existing Newsroom functionalities will be maintained.

It will not be possible to create layout that automatically clip the external page layouts. Newsroom elements added to the page will overlap the EPS representing the preview of the external page.

Information about added elements will be stored into the content field but not in the external pages markers, which will be skipped by Newsroom each time an operation is performed.

Complex pages with shapes of both applications (Newsroom and external application) will be created by overlapping the elements and no constraint check will be done by Newsroom. The external page will be a read only page.

4 Performance

The management of layered pages will not degrade the performances of NewsRoom. In fact, by using the external layout strategy, all the operations will be transparently handled by NewsRoom, without requiring code overhead to handle special cases for external pages.

4.1 Speed

No degradation in speed due to the implementation of the system.

4.2 Reliability, availability and maintainability

No degradation in reliability, availability and maintainability due to the implementation of the system.

4.3 Capacity

No degradation in capacity due to the implementation of the system.



Integration Platform Project Supplementary Specification

HE70-SS-IP.doc

**Authorized for issue
by:**

G Colagrossi

**Unisys
Integration Plan
Project Manager**

Signature

Date

A Politi

**Unisys
Integration Platform
and Adobe
Integration
Project Manager**

Signature

Date

**December 11, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Integration Platform Project Supplementary Specification				
Author	A Politi Adobe Integration and Integration Platform Project Manager Unisys				
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys				
Comments	This is the Supplementary Specification for the Integration Platform project				
File Name	HE70-SS-IP.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	12/11/2002	BC	First version of the Functional Specification

Table of Contents

1	INTRODUCTION	5
2	OVERVIEW	5
3	TECHNICAL REQUIREMENT	7
4	ADVANTAGES	7
5	CONCLUSION.....	8

1 Introduction

The Hermes Integration Platform is a new development that enables third party applications to be part of the Hermes editorial system by using the standard protocol over the HTTP transport protocol.

The following document shows a general picture of the currently implemented software prototype that has been shown at the recent IFRA Expo and a proposed implementation for the final product.

2 Overview

The current implementation of the SOAP Integration Framework relies on the HermesAgent service which handles the incoming HTTP requests.

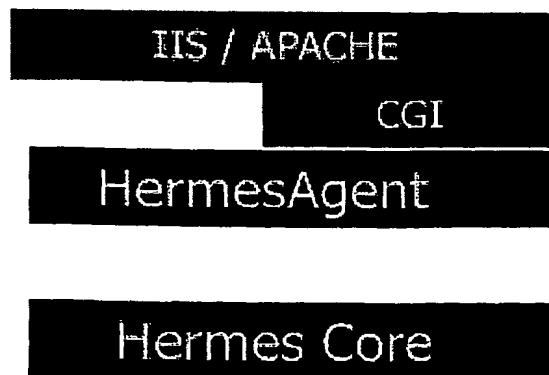
Each request is verified against the presence of a particular HTTP header value (SOAPAction) that, if the header value is set, copies the content of the request to the SOAP core library.

The SOAP core library is responsible for parsing the XML request, determining which method has been called via SOAP and dispatching the method call to the appropriate WEB service. The WEB services are loaded dynamically on demand and are implemented as a separated piece of running units.

HermesAgent is a module acting as a bridge between the WEB world and the Hermes system by taking care, for example, of dispatching sessions and managing the Hermes configuration in a environment that is slightly different from the Hermes usual one.

The HermesAgent has not been created as a replacement of a WEB server but to be the easiest and quick deliverable way to enable Hermes to the WEB. Thus, the overall architecture is comprehensive of a standard Web Server (IIS, Apache), which handles the connections from the WEB and calls a "plugin" that runs on the HermesAgent via a CGI application.

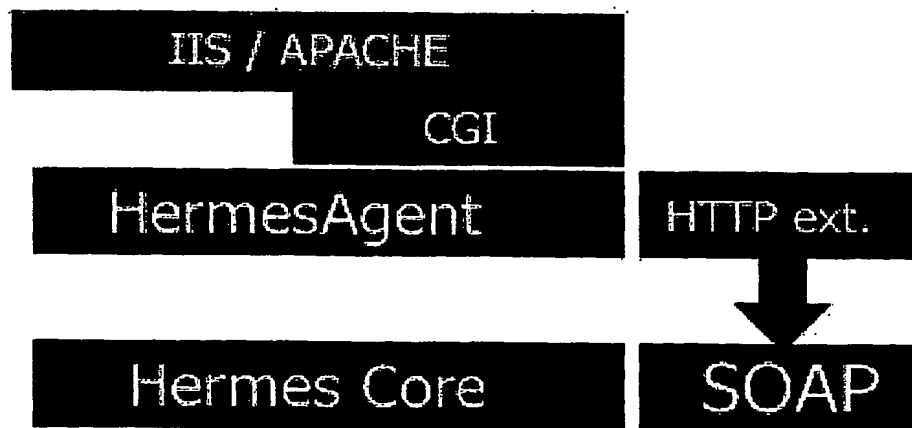
The overall picture is shown below



Due to a serious issue arisen with IIS and the Hermes System together, the HermesAgent has been extended for the IFRA Expo prototype to directly support the HTTP protocol without using a CGI.

The extension has been made not to replace a complete commercial Web Server but to overcome the cross-platform issues between different Web Server implementations. In fact, CGI would not be used as a pass through modules because the performance issues and the server-based modules are based on proprietary and different technologies such as Microsoft ISAPI.

Thus, the solution used for the prototype has been modified as shown below:



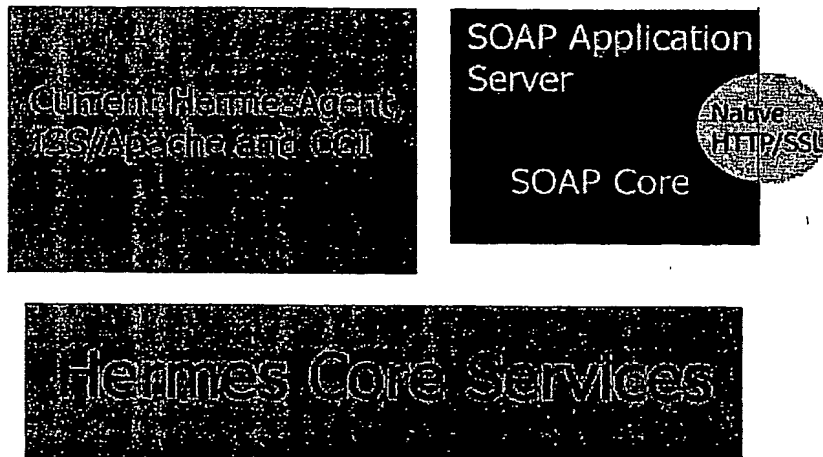
The current working solution is not a deliverable product since it has been built for the prototype purpose only, therefore without all the care required by a real product.

However, extending the HermesAgent to fulfill the requirement of the SOAP Integration Platform can be dangerous since it is currently used in the already released HermesWeb, which uses for the internal networking communication the Unisys proprietary protocol. Adding a new communication protocol could lead to a mixed, hybrid solution which could slow down the overall performances of both usage scenarios of the Hermes Agent. Furthermore, one of the key requirements of the Integration Platform is the secure connection implementation (HTTPS) and implementing it in the current HermesAgent can be a hard task because the compatibility with the currently production software must be maintained.

After the IFRA expo, the R&D team has analyzed the HermesAgent prototype implementation and found that most of the code could be moved in a small framework which supports HTTP/HTTPS directly, without interacting with a web server.

3 Technical Requirement

With the decision to release the Integration Platform with Hermes 7.0 (internal name), there will be the opportunity to leverage the existing HTTP framework used in the prototype and make a new service that will be called SOAP Server. In this way, the Integration Platform can become a complete and separated licensed product. The requirements that the new service must have are listed here below.



HTTP and HTTPS support

The new service must support HTTP connection as well as HTTP with SSL.

Performances

The new service must be able to support concurrent connections

Hermes core reuse

The SOAP service will rely on the existing Hermes Server Framework to take advantage of the proven stability of the APIs.

4 Advantages

Having a separate complete independent SOAP Integration platform is the preferred solution for the following reasons:

1. Software

- a. A different code limits the impact of the existing production one.
- b. The performances can be dramatically improved because it is not necessary to mediate between what has been done for the WEB and what is being doing for the SOAP Integration Platform anymore. Typically, the amount of data passed through the Hermes Web via the HermesAgent is not very high. Conversely,

the SOAP Integration Platform manages native files created by different applications and EPS that can be very huge. Using the same Web interface for both Hermes Web and SOAP Platform can result in a serious degradation of performance and it is expected that, when two Adobe clients will send two EPS of tenth of MB in size, no other can work friendly with Hermes Web.

- c. No more needs to maintain a timeout on the session (which is required for the Web). It is possible to implement the same behavior as in Hermes and this is a great improvement. Suppose that a huge InDesign file needs to be released to Hermes. With the current solution, embedded in the HermesAgent, after a period of time the session is closed and no more operations can be performed until a new login is made. But what if the timeout is passed because the InDesign file is still being converted into a very huge EPS. When InDesign finishes the EPS build, it is ready to send data, but the server closes the session. Also the setting of a very high timeout value does not solve the problem. Note also that this is a "must have" during the currently performed demos.

2. Commercial reason

Unisys will release the Integration Platform with a separate licensing policy and with a different price.

5 Conclusion

It has been pointed out that the Hermes Integration Platform needs an HTTP service to work and the solution proposed at IFRA 2002 based on the HermesAgent has produced some important results. Furthermore, the initial requirement was to reuse an existing Hermes Web implementation.



Database Format Field

Project Functional Specification

HE70-FS-DFF.doc

**Authorized for issue
by:**

U Gadarco

Unisys Project Manager

Signature

Date

H Hofmann

**Unisys
Editorial Product
Manager**

Signature

Date

December 11, 2002
Version 1.0



Unisys Proprietary Data Notice


The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

**Copyright © 2002 Unisys Corporation
All rights reserved**



Project Information

Document Title	Database Format Field Project Functional Specification				
Author:	BC Requirements Analyst Unisys				
To Be Approved By	UG Project Manager Unisys	HH Editorial Product Manager Unisys			
Comments	This is the Functional Specification for the Database Format Field project				
File Name	HE70-FS-DFF.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	12/11/2002	BC	First version of the Functional Specification

Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	5
2.3	ASSUMPTIONS AND DEPENDENCIES	5
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE	5
3	FUNCTIONAL DESCRIPTION.....	6
3.1	THE DATABASE FORMAT FIELD.....	6
3.1.1	Description.....	6
4	INFORMATION FOR USER MANUAL	7
5	PERFORMANCE	7
5.1	SPEED.....	7
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY.....	7
5.3	CAPACITY.....	7
6	INSTALLATION	7



1 References

This functional specification refers to Database Format Field 3PI project.

2 Functional Overview

In order to store information relevant to third party native applications of objects/pages to be retrieved from the Hermes database, a string field will be added to the pages table of the Hermes database.

2.1 Major functions

Availability of a string field in the Hermes database, where the mime type of the application that creates the external page/content will be stored.

2.2 Data/Activity Diagram

Not applicable.

2.3 Assumptions and Dependencies

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.

New functionality must remain compatible with prior versions of software.

3 Functional Description

Third party applications store their content in Hermes by placing the native format file onto the file system. In order to allow Hermes clients to show the content of pages and objects created by third party applications, a preview of the page/object is generated and stored in Hermes along with the native file.

If required, the native file can be reloaded in the originating application in order to be modified.

In order to store information able to identify the third parties native applications so that, when retrieving object/pages from the Hermes database with Hermes Explorer this information is shown in the Browse pane, a string field will be added to the pages table in the DB.

3.1 The Database Format Field

3.1.1 Description

A string field will be added to the pages table of the Hermes database to store the mime type of the application that creates the external page/object.

The mime type will be used to uniquely identify the external application. This will allow to have this information (object/page origin) displayed in the format fields of the Hermes Explorer Browse pane when retrieving pages.



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.

UNISYS

Adobe® Integration with eEditorial® Solutions Project Design Specification

HE70-DS-AIES.doc

Authorized for issue
by:

G Colagrossi

Unisys
Integration Plan
Project Manager

Signature

Date

A Politi

Unisys
Adobe Integration
and Integration
Platform
Project Manager

Signature

Date

October 15, 2002
Version 1.0



Unisys Proprietary Data Notice


The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

**Copyright © 2002 Unisys Corporation
All rights reserved**



Project Information

Document Title	Adobe Integration with eEditorial Solutions Project Design Specification				
Author	A. Politi Adobe Integration and Integration Platform Project Manager Unisys				
To be approved by	AP Adobe Integration and Integration Platform Project Manager Unisys				
Comments	This is the Software Specification for the Adobe Integration with eEditorial Solutions project				
File Name	HE70-DS-AIES.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	10/15/2002	AP	First version of the Design Specification
1.0	All	Update	12/12/2002	AP	
1.0	All	Update		AP	<ul style="list-style-type: none">• Changed SOAPAction values for login/logout and normal operations• Added section on metadata with XMP• Removed unnecessary prefix "hem" on each tag for request/response

Table of Contents

1	INTRODUCTION.....	6
2	REFERENCES.....	6
3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS	7
4	DOCUMENT CONVENTIONS	7
5	ABSTRACT	7
6	MOTIVATION.....	8
7	INTRODUCTORY STATEMENT.....	8
8	OVERALL ARCHITECTURE.....	8
9	GENERAL CONCEPTS	9
9.1	REQUESTS	9
9.2	RESPONSE FROM THE SERVER	11
9.3	QUICK INTRODUCTION TO THE HERMES LEVELS	12
9.4	MESSAGE WITH ATTACHMENTS	13
10	INTEGRATION PLATFORM SERVICES	15
10.1	SECURITY SERVICES	15
10.1.1	Login	15
10.1.2	Logout	17
10.2	OBJECT SERVICES.....	17
10.2.1	Object creation process.....	17
10.2.2	Placing objects into pages.....	19
10.2.3	Saving objects	20
10.2.4	Loading Objects	20
10.2.5	Delete Object.....	21
10.3	VERSIONING	21
10.4	PAGE SERVICES	22
10.4.1	Logical pages	22
10.4.2	Page creation process	22
10.4.3	Saving Pages	23
10.4.4	Linking Object to pages	24
10.4.5	Unlinking Object from pages	24
10.4.6	Delete Page	24
10.4.7	Page grids	24
10.4.7.1	Local page grids	25
10.4.8	Multiple pages	25
10.5	EDITION SERVICES	26
10.6	QUERY SERVICES	26
10.6.1	Page query	27
10.6.1.1	General considerations.....	27
10.6.1.2	User Interface.....	27
10.6.1.3	Level selection	27
10.6.1.4	Include sublevels.....	28
10.6.1.5	Edition	28
10.6.1.6	Publication Date.....	28
10.6.1.7	Page being modified	28
10.6.1.8	Workflow	29
10.6.1.9	Creator	29
10.6.1.10	Creation date range and Modification date range.....	30
10.6.1.11	Metadata search	30
11	CONCURRENT OPERATIONS.....	31

12	EVENT MANAGEMENT SERVICES	32
13	MESSAGING	32
14	SECURE HTTP.....	32
15	MANAGED ASSET TYPES	32
16	MOVING / COPYING OBJECTS AND PAGES.....	33
17	METADATA	33
18	PLUGIN SPECIFIC FUNCTIONALITIES	34
18.1	DUMMY TEXT.....	34
18.2	TEXT CONVERSION BETWEEN INCOPY AND NEWSROOM	34
I.	APPENDIX A: FAULT CODES.....	35
II.	APPENDIX B - USER INTERFACE	36
A.	QUERY OBJECT OPTIONS DIALOG.....	36
B.	QUERY PAGES OPTIONS.....	37
C.	RELEASE PAGE/OBJECT.....	38
D.	OBJECT HISTORY	39

1 Introduction

This document describes the new Hermes WEB Services API developed to integrate 3rd party components and acts as a programmers guide for the Integration Platform.

Although this document is subject to changes since the API specification is still under development, the basic infrastructure concepts here described will remain valid and can be used for building a prototype of the integration.

This document deals with the process of creating, saving, releasing and loading pages between Adobe InDesign, InCopy objects and e-Editorial News Content Manager.

2 References

Table 1 Referenced specification and standards

Description	Reference
SOAP Specification version 1.3	http://w3.org/soap
OpenSSL specification and downloads	http://www.openssl.org
Multipart messages	http://ietf.org/rfc.html RFC 2387
IANA official registered mime types	http://www.iana.org
OPI 1.3 specification	http://partners.adobe.com/asn/developer/pdfs/tn/OPI_13.pdf
OPI 2.0 Specification	http://partners.adobe.com/asn/developer/pdfs/tn/5660.OPI_2.0.pdf
Adobe XMP technology specification	http://partners.adobe.com/
The Unified Modelling Language specification version 2.0	http://www.rational.com

3 Definitions, Acronyms and Abbreviations

Acronym	Definition
IPC	Inter Process Communication
WSDL	Web Services Definition Language
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
HTTP	Hyper Text Transmission Protocol
HTTP-S	Hyper Text Transmission Protocol – with data encryption
SSL	Secure Socket Layer
XML	Extensible Markup Language

4 Document Conventions

The following typographical characters are used to identify the examples shown in this document:

courier bold

elements written in courier bold indicate code fragments

5 Abstract

The API implements a subset of the SOAP specification.

The WSDL support is currently under evaluation and, if required, it will be introduced in further developments.

For a complete discussion about RPC with SOAP, see the reference table.

6 Motivation

The creation of this new API is due to the need to make Adobe applications more tightly integrated with the Unisys e-Editorial Applications.

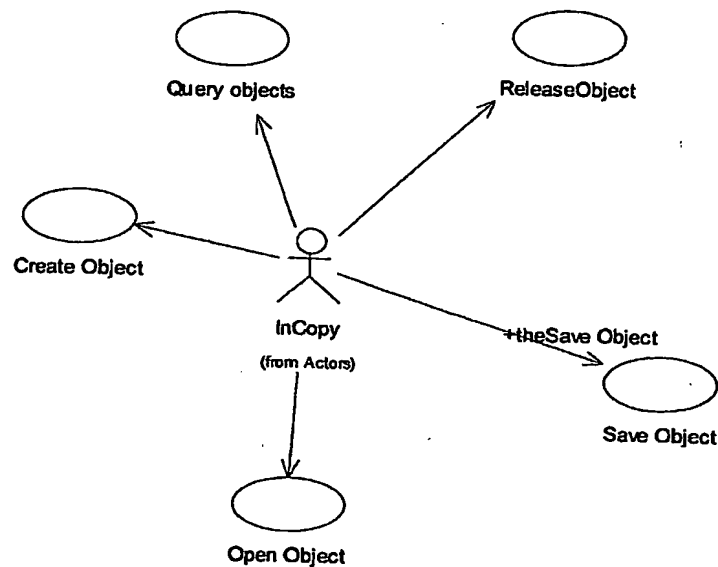
7 Introductory statement

eEditorial saves InDesign documents as logical pages and InCopy objects as "external objects". In order to complete the pagination within the Hermes editorial system and to be tightly integrated with it, Hermes SOAP framework will provide a set of API to allow InDesign to create logical pages and InCopy to create objects in News Content Management.

8 Overall architecture

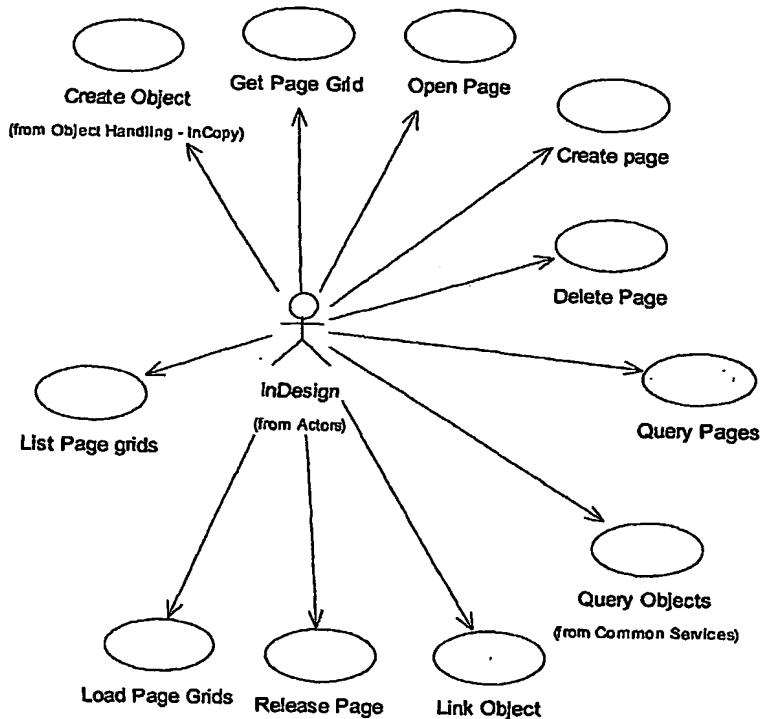
Adobe InDesign and InCopy will be extended with plug-ins to perform a complete interaction with Hermes Editorial System. Following is a high-level use case diagram relevant to the InCopy integration.

Figure 1 High-level use case diagram for InCopy integration



Adobe InDesign is extended to support operation on pages. A high-level use case diagram for InDesign integration is provided below

Figure 3 - High-level use case diagram for InDesign integration



9 General concepts

Attached to this document is the XML schema of the data types, parameters and method calls. Readers must refer to the schema file in order to correctly identify their representation.

e-Editorial SOAP methods are mapped to the **hem** namespace, thus each method name will have the "hem" prefix, as well as the response.

9.1 Requests

A request is carried out by an HTTP message that contains, in the header, the **SOAPAction** attribute.

The SOAPAction can have three different values, as depicted in Table 3

A typical HTTP header for a request is provided below:

```
POST <uri> HTTP/1.1
Host: XXXXX
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: Hermessoap
SESSIONID: <session_id>
```

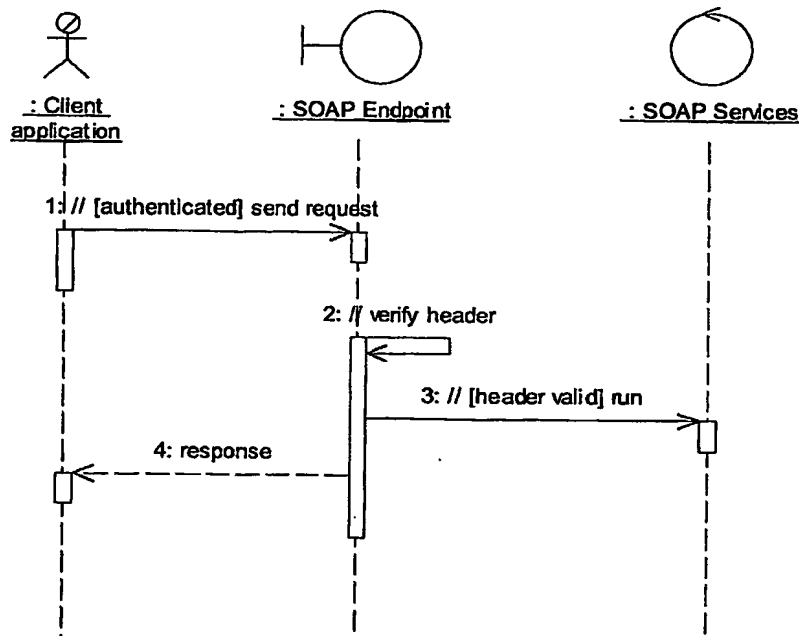
Table 3 Values for SOAPAction HTTP Header element

Value of SOAPAction	Purpose
HermessoapLogin	This value must be used by client when accessing the security services for a login.
Hermessoap	This value must be used by client when accessing every service different from login/logout
HermessoapLogout	This value must be used by client when accessing the security services for a logout.

The SESSIONID is obtained during the login phase in response to a successful login, and must be written in the HTTP header of each method call.

Client application can leverage the common login service by asking to the SOAP server for a previously created SESSIONID. Two applications running on the same machine will have a single login point. SOAP requests without the SESSIONID header attribute will be discarded without providing any error messages or notifications to the client. The sequence diagram for the overall request is shown in

Figure 5 - Sequence diagram for a SOAP request/response



9.2 Response from the server

Responses to an e-Editorial SOAP server method call is composed of 2 sections of the HTTP message.

The HTTP header contains the HTTP status code, while the body of the message will contain the response, which will be either a result or a SOAP fault.

Method response elements are "encoded" as per the W3C recommendation, appending the **Response** word to the called method name. For example, the response to the **HermesLogin** will be **HermesLoginResponse**.

Whenever a fault is returned to the caller application, the HTTP header will contain **HTTP/1.1 200 OK** and the body of the message will contain the actual fault structured as:

```
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
```

```
<faultcode>SOAP-ENV:"Error"</faultcode>
<faultstring>Description</faultstring>
<detail>
  <message> Hermes Error Message</message>
  <code> Hermes Error Code</code>
</detail>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
```

Faultcode and faultstring contain respectively the SOAP error code and the SOAP error message, as per SOAP specification, and are shown in the following table.

The choice of returning a 200 OK header is due to a mismatch between the specifications of HTTP protocol and the SOAP. As per specification, the HTTP header 500 Internal Server Error forces clients to ignore the body content while the SOAP specification uses the body to return the SOAP Fault

Table 5 - SOAP Fault codes

Fault Code	Fault String	Description
100	Version Mismatch	The call was using an unsupported SOAP version.
200	Must Understand	An XML element that was not understood by the receiver contained an element tagged with mustUnderstand="true".
300	Invalid Request	The receiving application did not process the request because it was incorrectly formed or not supported by the application.
400	Application Faulted	The receiving application faulted when processing the request. The detail element contains the application-specific fault.

Appendix A shows the list of Hermes error code and the relevant error messages.

The SOAP Fault error message element must be used to display a warning/error message to the user. The dialog window will display the error string and, only if a specific debugging option is set, the error code, used for debugging purpose.

9.3 Quick introduction to the Hermes Levels

In this document the concept of *levels* is frequently used. The Hermes level is identified by a unique number composed of a five bytes in a dotted format and represents the level id.

Hermes levels are comparable to a file system structure. Each level is thus specified by a name, which is the full path from the root level composed of

the concatenation of level names separated by a separator char, typically a slash.

As in the file system, each level has a set of permissions that enable or disable functionalities and the possibility of saving, creating or reading objects or pages.

Another important attribute is the level *type*. Although Hermes uses several level types to reference different object types, the level types that should be taken into account in this integration are PAGE LEVELS and OBJECT LEVELS, respectively enabled to accept logical pages and objects. Based on the Hermes configuration, a user can have a default level. If a default level has been configured for a specific user, the dialog box that shows level names must display the default level name on opening.

The type of level is identified by the `LevelType` element in the schema file. Most of the SOAP APIs require a level to be specified, otherwise in particular cases, SOAP client applications must supply the LEVEL ID, while their user interface must show the level name to the end-user.

The maximum visible depth of Hermes Levels should be 5.

9.4 Message with attachments

The body of such a message does not differ from other types of messages except for the fact that it is a multipart MIME message and the MIME blocks are referenced inside the parameters of the method call.

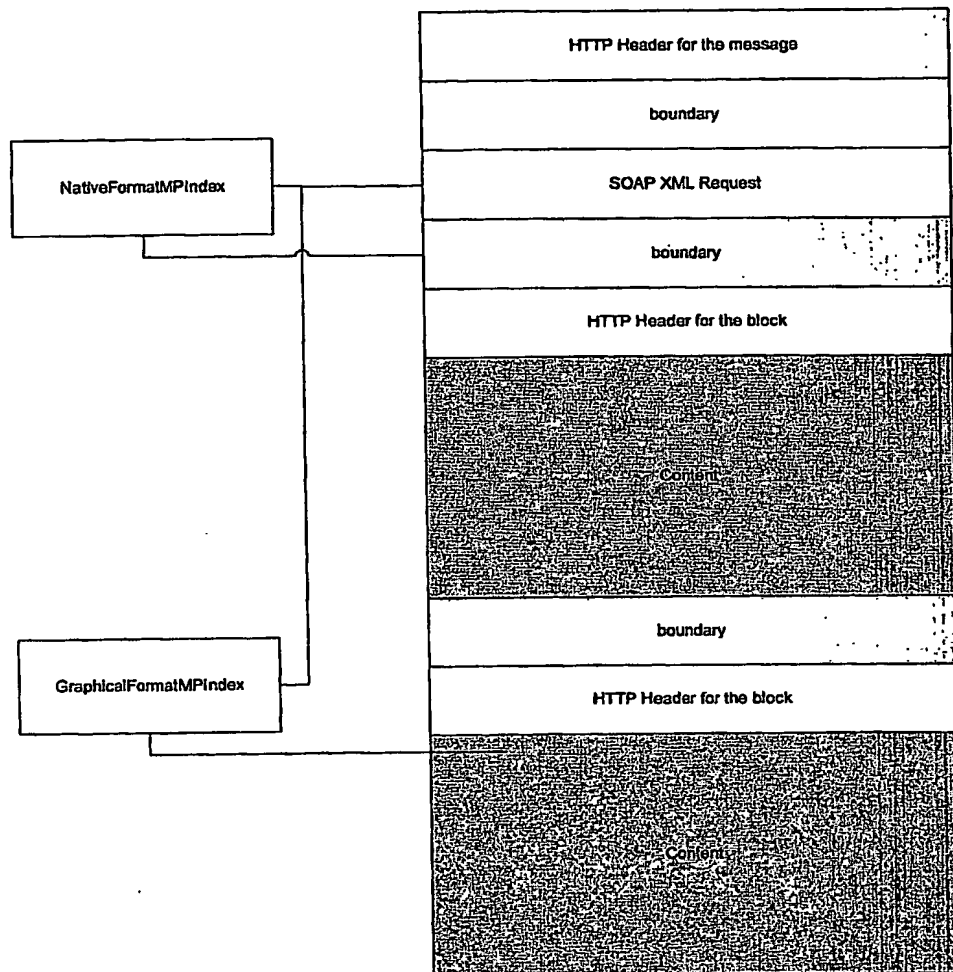
The W3C proposed standard for the Message with Attachments does not give constraints about the implementation. In this document, an easy to implement strategy is proposed.

When a call to a method that requires the presence of the native file format and/or the EPS is performed, there must be always a tag that is used to reference the bytes block inside the message content.

As depicted in the XML schema, whenever a file is exchanged back and forth between the Editorial SOAP server and InDesign/InCopy, the relevant method call (or the result of a call) will contain an element `<XXXFormatMPIndex>` with the content equal to the index of the message part that will contain that file.

An example of a call to save the InDesign page with the `HermesSavePage` method call is provided below.

Figure 7 - Message with attachments



To correctly locate the referenced part in the multipart mime, the content of the `<XXXXFormatMPIndex>` tag is used as if it were the one-based subscript of an array of boundaries. The very first boundary (which is the number zero) will be the SOAP message body.

The sample above references the 1st boundary in the HTTP message. An example of message with upload request attachment is:

```
<hem:HermesSaveObject>
  <NativeFormatMPIndex>1</NativeFormatMPIndex>
  <GraphicalFormatMPIndex>2</GraphicalFormatMPIndex>
</hem:HermesSaveObject>
```

The rest of the multipart MIME message will proceed as specified in the relevant W3C specification, with the file name in the header of the block and the block enclosed by boundaries. The same message type is used in response of `HermesGetObject` call.

Note: to protect itself from the potential proliferation of EPS, the e-Editorial server will associate persistently the name of the EPS sent during a `HermesSaveObject` (release). Thus, subsequent calls to `HermesSaveObject` in release mode must send the EPS with the same name used during the first upload. If different graphical attachments are attached to the message, they must be specified in the message body with the `<AdditionalAttachment>` XML tag. The value of this tag will contain the index to the block containing the actual attachment.

Unisys is currently considering to adopt the preliminary specification of the message with attachments from W3C. Currently, the specification is a proposed standard and the ability of the most used XML parser to accept these types of messages is under evaluation.

Messages with attachments are, in the new specification, XML messages that are not sent as multipart-mime and that contain the binary part of the attachment inside the XML.

10 Integration Platform Services

This section of the document explains the APIs to interact with services provided by the SOAP Integration Platform

10.1 Security Services

Security services provide a set of methods to interact with the Hermes Security subsystem

10.1.1 Login

While the login process remains unchanged compared to the previous integration architecture, the request will be made by calling the `HermesLogin` API and passing the same parameter values as the previous request. The login request is described in the schema file.

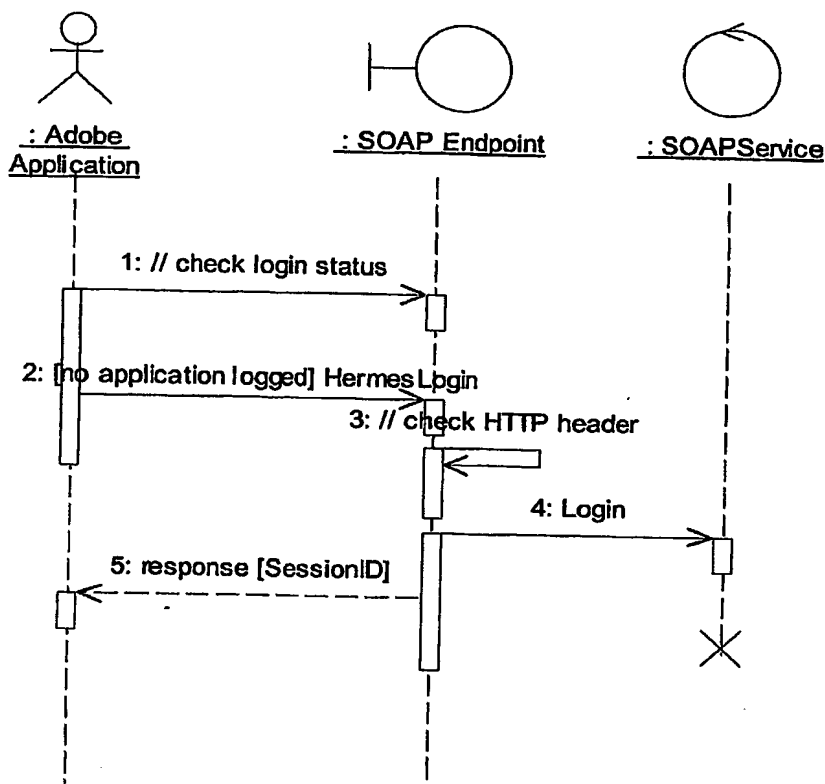
Once the client application has been authenticated by the server, it receives the `sessionId` in a Base64 encoded format, as in the previous version.

The session id, in this new integration architecture, will be referenced by the client by setting the HTTP header `SESSIONID`.

For each client message request, the HTTP header will specify the `SESSIONID`.

```
POST /<uri> HTTP/1.1
Host: XXXXX
Content-Type: text/xml; charset="utf-8"
Content-Length: nnnn
SOAPAction: HermesSOAPLogin
SESSION_ID: 261663
```


Figure 9 Login request - Normal flow



The response to a **HermesLogin** call contains the sessionID and timeout. Login response will be:

```

<SOAP-ENV:Body>
  <hem:HermesLoginResponse>
    <SessionID>129392</SessionID>
    <Timeout>100</Timeout>
  </hem:HermesLoginResponse>
</SOAP-ENV:Body>

```

The timeout value must be used by the client application to determine how long a session will remain valid and will accept method calls.

If the user cannot be authenticated by the e-Editorial SOAP server, the response will be a SOAP-Fault.

By leveraging the common login service, an application must check the status of authentication on a workstation. If one application has been previously authenticated by the SOAP server, other applications can obtain a valid session id by calling the **HermesIsLogged** SOAP message.

10.1.2 Logout

To logout from the Hermes SOAP server, client application must call the `HermesLogout` SOAP API and pass the `sessionid` as parameters of the message.

The HTTP SOAPAction header entry must contain the `HermesSOAPLogout` value.

10.2 Object Services

The following section describes the set of operations that can be performed on objects.

10.2.1 Object creation process

When the client application needs to create a new object inside the Hermes News Content Manager, it calls the `HermesCreateObject` method and passes the required parameters. Before doing so, the client application displays a dialog box asking the user to enter at least the following data:

- Level
- Edition
- Object name
- (PubDate)

After selecting the level from the list obtained by the `HermesEnumLevels` method call, the client calls the `HermesEnumEditions` and passes the selected level to obtain the list of available editions. If the selected level requires a publication date (see `PubDate` data type definition for details), the dialog box must require the user to supply a publication date, which will be verified later by the server.

The client application will supply the type of object as a string identifying the MIME type of the application that generates the object.

If the object can be successfully created, the SOAP server will return a `HTTP 200 OK` response and the body will contain the initial status of the newly created object.

The status will be referenced in subsequent calls, for example in the call to `HermesEnumNextValidStatuses`.

```
HTTP/1.1 200 OK
-
<SOAP-ENV:Body>
  <hem:HermesCreateObjectResponse>
```

```
<StatusID>12</ StatusID>
<ObjectID>73774</ObjectID>
</hem:HermesCreateObjectResponse>
</SOAP-ENV:Body>
```

If the object cannot be created, a SOAP fault will be returned.

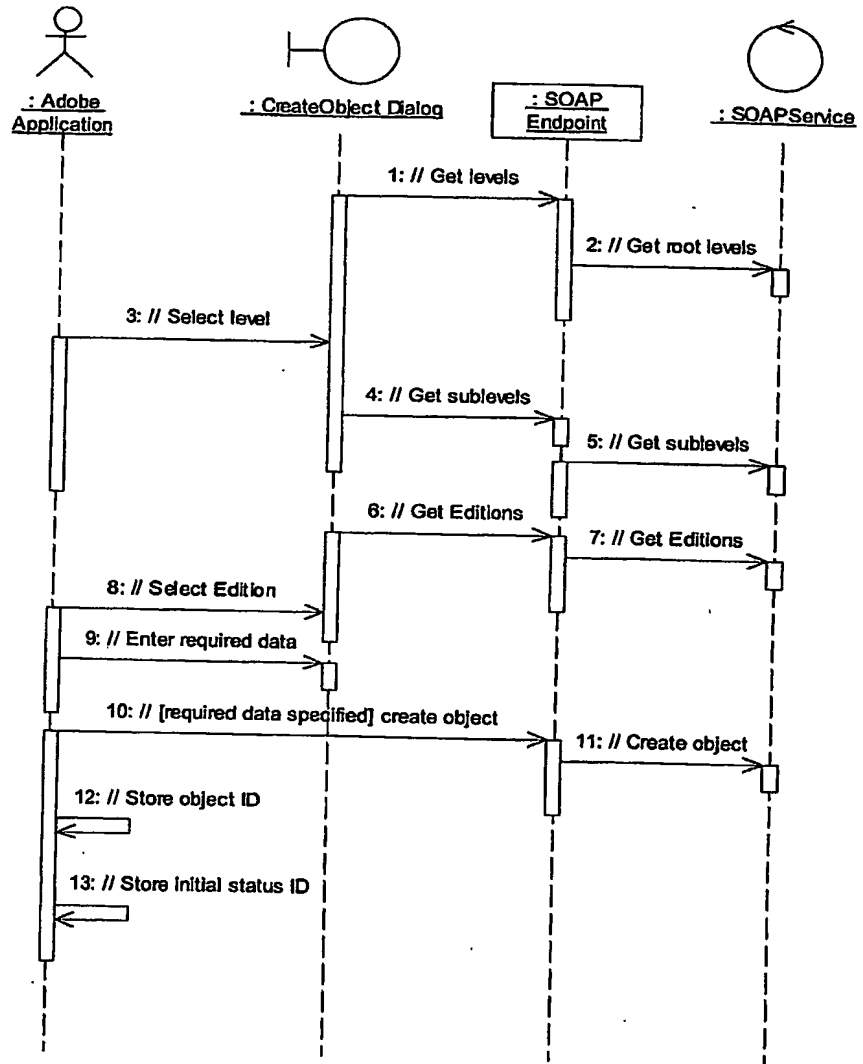
The Hermes eEditorial server locks the object entry to prevent another application from changing its data and metadata until a **HermesSaveObject** method call (or an explicit **HermesUnlockObject**) is performed with the flag **UnlockAfter** set to true. (In this case, the client must supply the native document).

Note that if the parameter of the **HermesCreateObject Unlock** is set to true, e-Editorial content server unlocks the object immediately after its creation. This can be useful for batch creation processes that need to create several objects entries without operating on them. For interactive applications, unlocking the page after the creation means that *the user cannot save the object* until it is locked again.

Tracking applications can monitor the activities on the allocated object also if it is locked, thus the metadata are accessible in read-only mode.

The application must cache the object unique identifier returned by the method call in order to use it in future operations such as **HermesSaveObject** and **HermesReleaseObject** to reference the object that is being processed.

Figure 11 Object creation overall sequence diagram



10.2.2 Placing objects into pages

Objects can be placed into pages only if they have been previously stored in the e-Editorial content management system.

Local files cannot be placed and saved into the NCM all over again. To prevent from doing so, the e-Editorial server will discard objects attached to a message that are not already inside the storage system.

In order to place a local file, the client application must call the `HermesCreateObject` SOAP API first. After a successful response from the Hermes SOAP Server, the object can be placed into the page. This is to allow Hermes Editorial system to keep track of objects usage and for further accountability on them.

The sequence diagram for the object placement is shown below.

10.2.3 Saving objects

The process of saving objects is carried out by the `HermesSaveObject` message. When saving it, the object will not be unlocked unless it is specified or the `UnlockAfter` element value is set equal to "true". If the page is being saved and released, the user must specify the status in which the page has to be released. The available statuses can be obtained by calling the `HermesEnumNextStatuses` API, which returns the valid status the object can be moved to according to the e-Editorial workflow subsystem.

The process of "saving" or "saving and releasing" objects requires the native InDesign/InCopy document to be sent by the client application.

Moreover, if a release action is being taken, the client application is responsible for creating and attaching the relevant EPS¹ document if required by the workflow. This means that if the user is releasing an object/page, the client application must check if the `ExtendedStatus` element is equal to "READY FOR TYPESET". To obtain status information, the client application must call the `HermesGetStatusData` API for the status that has been selected by the user.

The message type used when moving InDesign /EPS files back and forth is called a message with attachments and will be discussed in the following section.

10.2.4 Loading Objects

Opening an object can be achieved through a call to `HermesGetObject` with the appropriate parameters.

Typically an object is loaded after a query is performed, either a simple level query or a parameterized query.

Hermes Image Object Types

To retrieve an image from the Hermes database, the client request must not specify the `NativeFormat` but only the `GraphicalFormat`. This is due to the fact that the native format is used only to store and retrieve an external object that cannot be handled by Hermes directly. For images, they are of common types such as JPEG, TIFF which are to be intended as graphical format.

¹ For convenience, the EPS format is used to handle previews. Hermes SOAP server accepts also a PDF or JPEG as a preview. This format MUST be specified in the `Format` parameter of the `HermesCreateObject/Page` SOAP API. If performance issues will arise in using the EPS format, we can discuss with Woodwing if using a PDF instead will increase the performances

10.2.5 Delete Object

The object deletion is achieved by calling the `HermesDeleteObject` SOAP API. If the object is locked, the operation fails and a SOAP fault is returned.

Getting object information

Information about an object can be retrieved by calling `HermesGetObject` API without specifying the `NativeFormat` element in the SOAP request. The returned information set will not contain any attachment but only the meta-information of the object.

10.3 Versioning

Versioning applies only to objects. Client applications must be able to retrieve specific versions of objects. By using the `HermesGetObject` method, client applications can retrieve specific versions of an asset by specifying the version ID of the object in the `VersionID` parameter. This parameter is not mandatory. If it is not specified, the most recent version is retrieved.

Client applications must supply a GUI component to allow the user to choose the version through a menu item such as "Get version". The version list will be sorted in a descending order, or last-first.

Under normal operating conditions, the user interface will not ask the user to choose a version.

To obtain the history of an asset, the client application can call the `HermesObjectHistory` API

Previous version of a story can be opened in read only mode.

The client application must supply a UI dialog to create a new version. A new version of an object is created whenever the object is **released**. Refer to the `HermesSaveObject` API for details about the release operation.

10.4 Page Services

10.4.1 Logical pages

10.4.2 Page creation process

InDesign documents are treated as Hermes Logical Pages.

The creation of logical pages process, as well as their retrieval, is similar to the object creation process described in § 10.2.1

InDesign pages are created by calling the `HermesCreatePage` SOAP API.

When the client application needs to create a new page inside the Hermes News Content Manager, it calls the `HermesCreatePage` method and passes the required parameters. Before doing so, the client application displays a dialog box to ask the user to enter at least the following data:

- Level
- Edition
- Page name
- (PubDate)

Logical pages can be fetched from the Hermes server only by specifying the full qualified path of the page, which is:

- LevelID
- Edition
- Page name
- PubDate

The `PageID` element that is returned from most of the SOAP calls is used internally by Hermes to keep the links in the inheritance tree. Trying to load a page by specifying the sole `PageID` will result in a *page not found* error.

After selecting the level from the list obtained by the `HermesEnumLevels` method call, the client calls the `HermesEnumEditions` and passes the selected level to obtain the list of available editions. If the selected level requires a publication date (see `PubDate` data type definition for details), the dialog box must require the user to supply a publication date, which will be verified later by the server.

The client application will supply the type of object as a string identifying the MIME type of the application that generates the object.

If the page can be successfully created, the e-Editorial SOAP server will return a HTTP 200 Ok response and the body will contain the initial status of the newly created page. The status will be referenced in subsequent calls, for example in the call to `HermesGetNextValidStatus`

```
HTTP/1.1 200 Ok
-
<SOAP-ENV:Body>
  <hem:HermesCreatePageResponse>
    <hem:StatusID>12</hem:StatusID>
    <hem:ObjectID>73774</hem:ObjectID>
  </hem:HermesCreatePageResponse>
</SOAP-ENV:Body>
```

If the page cannot be created, a SOAP fault will be returned.

The e-Editorial server locks the page entry to prevent another application from changing its metadata until a **HermesSavePage** method call (or an explicit **HermesUnlockObject**) is performed with the flag **UnlockAfter** set to true. (In this case, the client must supply the native document).

Note that if the parameter of the **HermesCreatePage Unlock** is set to true, e-Editorial content server unlocks the page immediately after its creation. This can be useful for batch creation processes that need to create several pages entries without operating on them. For interactive applications, unlocking the page after the creation means that *the user cannot save the page* until it is locked again.

Tracking applications can monitor the activities on the allocated pages also if it is locked, thus the metadata are accessible in read-only mode.

The application must cache the page unique identifier returned by the method call in order to use it in future operations such as **HermesSavePage** and **HermesReleasePage** to reference the page that is being processed.

10.4.3 Saving Pages

The process of saving logical pages is carried out by the **HermesSavePage** message. When saving it, the page will not be unlocked unless it is specified or the **UnlockAfter** element values set equal to "true". If the page is being saved and released, the user must specify the status in which the page has to be released. The available statuses can be obtained by calling the **HermesEnumNextStatuses** API, which returns the valid status the object can be moved to according to the e-Editorial workflow subsystem.

The process of "saving" or "saving and releasing" pages requires the native InDesign/InCopy document to be sent by the client application.

At each save operation, client application should attach a JPEG preview in order to let Hermes display a preview in all other client applications.

Moreover, if a release action is being taken, the client application is responsible for creating and attaching the relevant EPS document if required by the workflow. This means that if the user is releasing a page, the client application must check if the **ExtendedStatus** element is equal to "READY FOR TYPESET". To obtain status information, the client application must call the **HermesGetStatusData** API for the status that has been selected by the user.

If the page contains links to images files that come from the Hermes system, OPI comments will be placed in the EPS to reference the path of the high-resolution version of the image. Hermes is compliant to the OPI standard version 1.3. Although OPI 2.0 is not a yet a market standard, OPI version 2.0 comments will be placed in the EPS together with the 1.3 version.

For the complete specification of OPI version 1.3 and 2.0, see the references table in this document

The message type used when moving InDesign /EPS files back and forth is called a message with attachments and will be discussed in chapter 1.1

10.4.4 Linking Object to pages

In order to allow Hermes to correctly manage the links between objects and pages (for example to allow queries such as “list all the objects that are linked to a specific page”), the client application must notify the linking operation by calling the **HermesLinkObject** SOAP API.

If the object cannot be linked to a page, for example when the object is already linked to another page, Hermes SOAP Server returns a SOAP Fault and the client application must inhibit the placing of objects.

To link object to pages, the **HermesLinkObject** API needs to know the pageid of the page in which the object is about to be placed. This implies that the page must exist in the database. Pages for which **HermesCreatePage** has not been already called must avoid the object linking.

10.4.5 Unlinking Object from pages

Whenever an object is unlinked from a page, the **HermesUnlinkObject** must be called. Both APIs need the full qualified page name element to be passed as parameters.

10.4.6 Delete Page

The page deletion is achieved by calling the **HermesDeletePage** SOAP API. If the page is locked, the operation fails and a SOAP fault is returned.

10.4.7 Page grids

Before the application can create a new Hermes page, the correct grid must be selected.

In Hermes, each level can have an attribute to indicate that a grid is available. If the attribute is not set, the grid is taken by the parent level. This algorithm stops at the root level and, if it does not have a grid, it will have a default grid.

At each level change, the client application queries for the grids available and presents the list to the user.

This process ensures that the page size is always synchronized with the page sizes defined onto the Hermes system.

To obtain the list of grids names, the client application must use the **HermesListGridNames** SOAP API call and specify the **LevelID** that the application is querying.

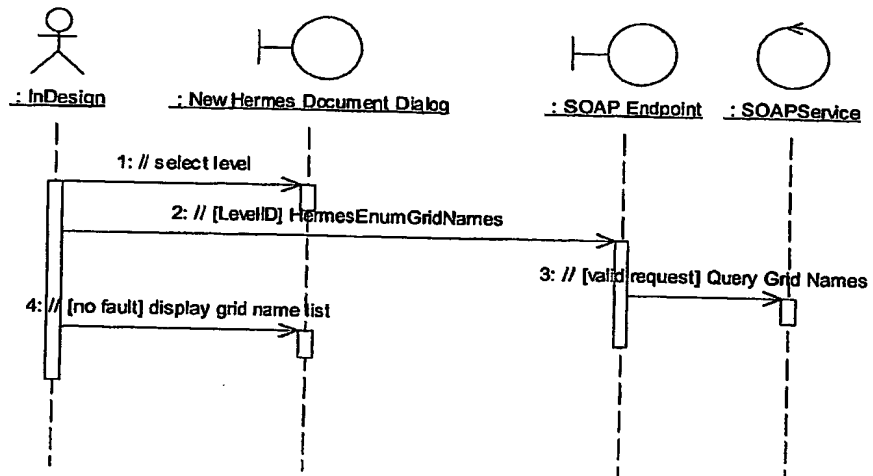
The actual grid loading will be possible by calling the **HermesLoadPageGrid** API, with **LevelID** and grid name as parameters.

A request for grid name listing will be

```
<hem:HermesEnumGridNames>
  <LevelID>1.1.0.0.0</LevelID>
</hem:HermesEnumGridNames>
```

shows the sequence diagram for the grid name listing operation.

Figure 13 Listing page grid names



The SOAP server response will be:

```
<hem:HermesEnumGridNamesResponse>
  <GridName>default</GridName>
  <GridName>layout</GridName>
</hem:HermesEnumGridNamesResponse>
```

Figure 13 shows the sequence diagram for the page grid loading.

10.4.7.1 Local page grids

To enable the user to work with InDesign and InCopy, page grids must be available also as a local cached copy. This will follow the previous implementation where grids were defined into a local XML file.

10.4.8 Multiple pages

If an InDesign document contains more than one page and it must be released in a status that requires the preview, the client applications must generate the preview for each page in a separate EPS/JPEG file. The entire package of file will then be saved using the normal procedures.

The message attachment for this document type will contain the InDesign native file and a series of blocks, one for each EPS.

10.5 Edition Services

Edition services provide access to edition information for a specific newspaper. By using the edition services provided by the Hermes SOAP server, client application can list edition names and obtain more specific information about a particular edition.

Editions are named and have a zone that enables the inheritance among them. For example, there could be an edition named A and an edition named A* that inherits from A. This means that pages inside edition A* inherit from pages in edition A.

10.6 Query Services

Query on pages/objects is performed through the `HermesQueryPages/HermesQueryObjects` call. When querying objects, user must be able to specify different parameters in order to refine the query and obtain only a list of objects that meet specific requisites.

Queries parameters are:

- Publication date from
- Publication date to. This value can be set only if the previous one has been entered
- Status
- Type of object: Images, InCopy, InDesign, Illustrator, etc (valid only for querying objects)
- Author
- Name of object / pages; wildcards are accepted. For the XML point of view, wildcards are included in the ANSI set of chars.
- Every parameter combination can be made and a query with all parameters specified can be issued.

The result set of a `HermesQueryObject/Page` call must be displayed with at least the following set of metadata:

- Object type
- Object Name
- Author
- Status
- Lock State

The display of the status information is up to the client implementation. It is suggested to draw the border of the item of the list containing the result with the color associated with the status of the object.

10.6.1 Page query

10.6.1.1 General considerations

The advanced query for page and objects allows the end user to perform sophisticated queries by specifying different parameters that will be evaluated by the SOAP server.

The GUI component is a three-tabbed dialog window.

10.6.1.2 User Interface

User will be able to specify different parameters for querying InDesign pages. Below is the screenshot depicting how the GUI component should be.

Figure 15 Page Name tab for page query

The screenshot shows a dialog box titled "Query pages" with three tabs: "Page Name", "User info", and "Page info". The "Page Name" tab is active. It contains several input fields and checkboxes. On the left, there are labels for "Level" (PUBLICATION, DEPARTMENT, ISSUE, LEVEL 4, LEVEL 5), "Edition", "Name", and "Publication Date". To the right of these labels are corresponding input fields. The "Level" field is set to "DEF_USER_LEVEL". The "DEPARTMENT" field is set to "COVER". The "ISSUE" field is set to "ARTS". The "LEVEL 4" field is set to "BIZ". The "LEVEL 5" field is set to "STORIES". There is a checkbox for "Include sublevels" which is unchecked. The "Edition" field is empty. The "Name" field is empty. The "Publication Date" checkbox is unchecked. Below the "Publication Date" checkbox are "From" and "To" date pickers. At the bottom is a "Page Comment" text area. On the right side of the dialog, there is a section titled "Page being modified" with checkboxes for "In use by" and "Not in use". Below this is a "Workflow" section with a list of roles: "HOLD", "EDITOR", and "REPORTER". At the bottom of the dialog are "OK", "Cancel", and "Apply" buttons.

10.6.1.3 Level selection

The main level selection is made into the SmartConnection panel. However, it should start with the default level (if any) defined for the user currently connected. To obtain this information, the application can call the `HermesGetUserData`

10.6.1.4 Include sublevels

This option is set via

```
<IncludeSublevels>true</IncludeSubLevels>
```

and enables the query to go through all the sublevel.

If the user specifies all the 5 levels by selecting values from the relevant combo boxes, this option will be hidden.

10.6.1.5 Edition

The combo box is filled with the edition names. This dataset is obtained by calling **HermesEnumEditions** SOAP method and specifying the level ID. For the query purpose the **EditionID** must always be used.

10.6.1.6 Publication Date

If this option is checked, the date from and to must be filled in. If the date to is not set, the current date will be used by default.

The fields will contain integers specifying valid dates number. At the moment, we assume that the date to be entered will have the following format:

MM DD YY

10.6.1.7 Page being modified

User can specify whether the query will search for pages currently being modified (locked) or not (unlocked).

Additionally, the user can specify the username of the user being modifying the page. This can be achieved by either entering the username manually or selecting the browse button.

Clicking the browse button will result in a call to **HermesEnumUsers** SOAP call and the results will be displayed in a list box into a separate dialog box. The list box will be single-selection.

This parameter is set with the **<Modifying>** tag in the request.

10.6.1.8 Workflow

It is always possible to specify the status of the pages to be searched, in order to filter the pages found in a different status. This list is multi-selection, thus allowing the user to select one or more statuses. In the query, this corresponds to a space-separated list values of the <StatusID>tag.

Figure 17 User Info tab for page query

Query pages

PageName User info Page info

Creator

Username

☐ Creation date range

From: [MM][DD][YYYY] To: [MM][DD][YYYY]

Modifier

Username

☐ Modification date range

From: [MM][DD][YYYY] To: [MM][DD][YYYY]

OK Cancel Apply

10.6.1.9 Creator

The username field corresponds to the <Creator> tag in the request and the behavior is the same as in previous page (*Page being modified* option).

10.6.1.10 Creation date range and Modification date range

This set of fields allows the user to specify the date/time from and date/time to that can be used for querying pages.

Figure 19 Page info tab for page query

The screenshot shows a 'Query pages' dialog box with three tabs: 'Page Name', 'User info', and 'Page info'. The 'Page info' tab is selected. It contains the following fields and input boxes:

- Title: []
- Author: []
- Description: []
- Job Name: []
- Additional Data: []
- Additional Data: []
- Additional Data: []

At the bottom of the dialog are three buttons: 'OK', 'Cancel', and 'Apply'.

10.6.1.11 Metadata search

This page is currently under definition because involves the XMP technology aspect and the ability to define custom fields and field prompts.

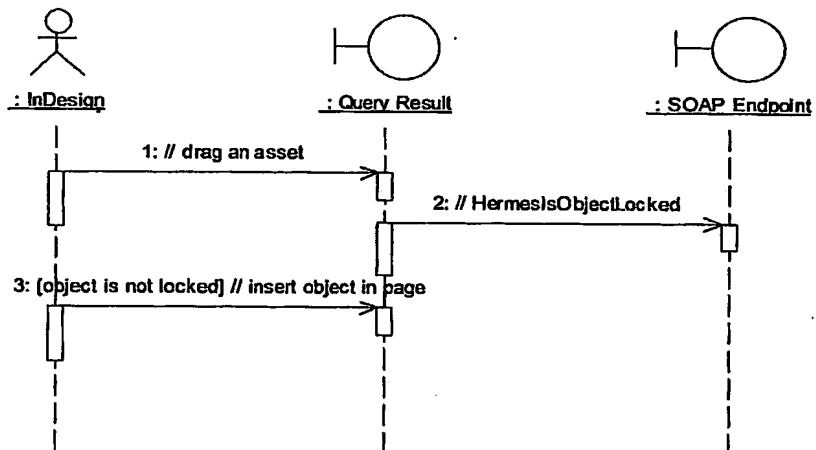


11 Concurrent operations

Hermes is a collaborative environment that allows multiple users to work on the same page. Client application can paginate or, generally, use objects only if they are not being used by others. Before the client application can make use of an object, it must query its Locking Status to verify that the object is not locked.

To verify the locking status of an object or page, client application can call the `HermesIsObjectLocked` (or `HermesIsPageLocked`) with the object/page ID of the object or page to be checked. An example could be:

Figure 21 Testing the lock status before adding objects to page



12 Event Management Services

Client applications must be compliant with the Hermes editorial system event management and tracking. Events are “notifications” about state changes and other operations made by clients into the Hermes system.

When a client receives an event it must update the information pertaining to the object or page the event is relevant to.

If a Hermes client changes the status of a page/object opened in InDesign / InCopy, ID/IC must update the status information.

Events are received from the common login application running on the client. Whenever an event is received, it is posted to the target application via a PostMessage on Windows and via the IPC (Inter Process Communication) on Macintosh.

13 Messaging

Hermes editorial system allows various clients on the network to collaborate via instant messaging, alerts and mail, the latter based on a private proprietary mailing system.

Client applications that needs to access the messaging service will call the common login via a shortcut.

14 Secure HTTP

In order to keep information safe and secure over the network, the Hermes environment uses the encryption on the proprietary protocol.

In order to achieve the same level of security, communication over the HTTP transport can be kept secure using the Secure Socket Layer.

Secure connections respond to the same endpoint on a different port. This allows clients to use both secure and unsecure connection to perform SOAP operations.

Unisys suggests using a freely available client library, called OpenSSL, which is recognized as to be a very good implementation of the HTTPS standard. OpenSSL is available at <http://www.openssl.org>

Clients using the LibWWW library to manage the HTTP connection can take advantage of the internal SSL implementation.

Unisys will not provide integrators with client certificates in order to use SSL over HTTP connection to perform SOAP interactions

15 Managed Asset Types

Hermes is capable of managing different types of application-specific objects. Different pages or objects can be generated by different type of applications resulting in a set of object/page types that can vary among configurations. A client application that is able to perform advanced query on the Hermes

database must be able to offer to the user the ability to select which type of asset to be queried, for example:

"List all the objects in a certain status and generated by InCopy and Illustrator"

To provide this "registry" information that can be used to show the list of managed asset types, Hermes SOAP server offers the `HermesEnumAssetTypes` API. The result is a list of application names/asset category pair which can be used to fill listboxes etc. Hermes SOAP will return all the managed asset types. The client application should filter this list to present only the assets that can be inserted or imported into the application.

16 Moving / Copying objects and pages

Client applications must be able to move and/or copy InDesign pages and InCopy object among levels. The SOAP API for copy does not differ from the one used to move an object/page except for the "Operation" parameter which must be equal to `copy` for a copy operation.

By default, the `HermesMoveObject/HermesMovePage` SOAP API moves the asset from one level / edition to another, deleting the content of the source object/page.

17 Metadata

For each type of object, the user must be able to change metadata such as the Author, content type, etc. XMP technology will be used for embedding metadata into InDesign pages or InCopy stories. Client application must supply a UI component to allow the user to change metadata.

The list of editable metadata is shown in the following table:

Table 7 Table of minimum required metadata

Attribute name	Attribute type	Applicability
Author name	String	Objects/pages
Comment	string	Objects/pages
Content description	String	Objects/pages

18 Plugin Specific Functionalities

18.1 Dummy Text

InCopy texts can be placed into the page regardless of the Hermes status they are associated with in the database. However, some statuses prevent the text from being viewed. In order to allow page designers working with filled text layout, InCopy object that are not in a particular status must be shown with a dummy text.

When the InCopy object is released in a status such as READY FOR TYPESET, the real text can be inserted in the relevant text frame.

The client application can query the current status "extended status" to search for the "visibility" of the content.

InCopy must deny the opening of a story which is not in the "Ready for typeset" status.

EPS and JPEG previews produced by InDesign will not reveal the content of a text story with dummy text.

18.2 Text conversion between InCopy and Newsroom

When a Hermes textual object such as a headline or an article is placed into an InDesign page or opened with InCopy, the client application will receive a flat text as part of the message response. The response to the HermesGetObject call will contain an attachment of type text. Depending on the action performed in the specific application, the object is linked or converted following the rules below:

When the user drags and drops a Newsroom textual object, InCopy will import a flat version of the Newsroom text, without markup commands and text styles and convert the text into a new InCopy object.

When the user drags and drops a Newsroom textual object into InDesign, the plugin should create a new InCopy object with the content read from the Newsroom object.

I. Appendix A: Fault codes

Table 9 Fault codes and messages

Code	Message
-12	Object not found
-13	Layout not found
-14	Page not found
-40	Level error
-41	Level Not Found
-42	Invalid Level Type
-43	Invalid Data
-150	Initial Status Error
-151	Status attribute error
-152	Status Error
-153	Status Not Found
-154	Undefined Status
-102	Timeout
-1000	Deadlock
-1001	Database Error
-1005	Query Syntax Error
-204	Invalid Location
-207	Not Logged In
-211	Invalid Edition ID
-212	Invalid Edition
-213	Invalid Master Edition
-214	Invalid Edition Time
-215	Invalid Edition Zone

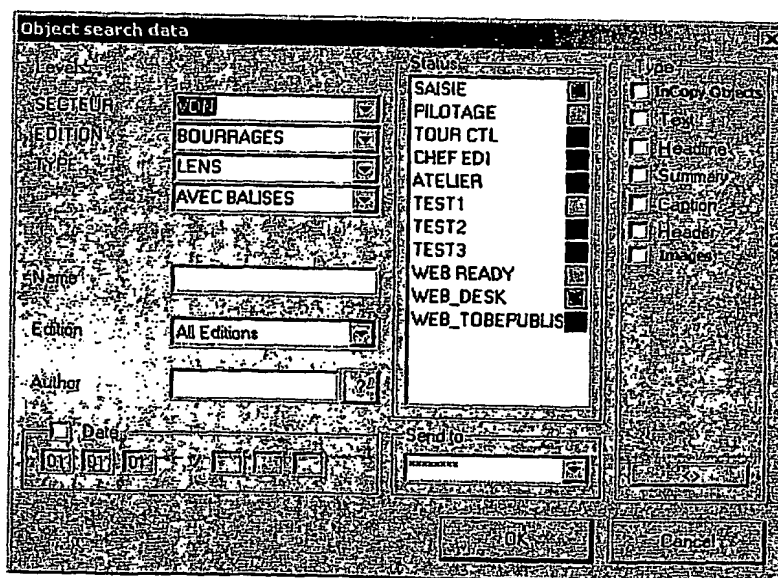
II. Appendix B - User Interface

This section shows some examples of GUI components used in the Hermes environment. Although the plug-in user interface shall be slightly different from the examples below, all the fields described must be used. Moreover Unisys suggests to dispose the fields as shown, in order to provide a way to work that is not very different from the Hermes one.

A. Query object options dialog

The query option dialog box is invoked by the client application whenever users want to specify additional criteria to retrieve objects from the Hermes News Content Manager repository.

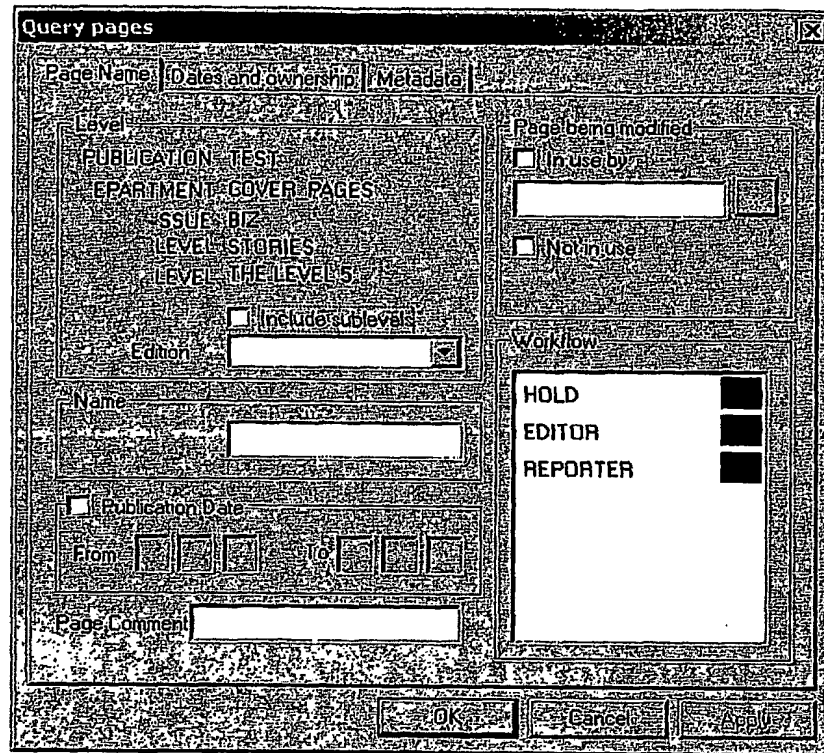
Figure 23 The Query Object option dialog box





B. Query Pages options

Figure 25 The Query Page option dialog box



The "Query pages" dialog box is a window with a title bar and a close button. It contains several tabs: "Page Name", "Dates and ownership", and "Metadata". The "Page Name" tab is selected. The dialog is divided into several sections:

- Level:** A list box containing "PUBLICATION TEST", "DEPARTMENT COVER PAGES", "ISSUE BIZ", "LEVEL STORIES", and "LEVEL THE LEVEL 5". Below this list is a checkbox labeled "Include sublevel".
- Page being modified:** A section with two checkboxes: "In use by" and "Not in use". Below these is a text input field.
- Workflow:** A section with three radio buttons labeled "HOLD", "EDITOR", and "REPORTER".
- Name:** A text input field.
- Publication Date:** A section with a checkbox labeled "Publication Date". Below this are two date pickers labeled "From" and "to".
- Page Comment:** A text input field.

At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

*Figure 26 Dates and ownership query dialog tab*

The 'Query pages' dialog box has three tabs: 'PageName', 'Dates and ownership', and 'Metadata'. The 'Dates and ownership' tab is active. It contains two sections for user selection and date ranges. The first section is for the 'Creator' with a 'Username' text field and a selection button. Below it is a checkbox for 'Creation date range' followed by 'From' and 'To' date pickers. The second section is for the 'Modifier' with a 'Username' text field and a selection button. Below it is a checkbox for 'Modification date range' followed by 'From' and 'To' date pickers. At the bottom are 'OK', 'Cancel', and 'Apply' buttons.

C. Release Page/Object



Figure 29 Sample release object - page dialog




The 'Release Page' dialog box contains the text 'Release the page in the following status'. Below this text is a list of three options, each with a square selection box: 'READY', 'MAKEUP', and 'READY FOR TYPESET'. At the bottom are 'OK' and 'Cancel' buttons.
















D. Object history







Argument under specification.





Icon Name	Current Icon	Description	Icon Group	Roll-over/Tool Tip Y/N
Newsroom		NewsRoom is the core of the system, it has the following main functional areas: Text Composition: an advanced tool for text and headline editing, provides the WYSIWYG view of articles, has a draft mode that displays text in a default style with or without typographical commands Page Layout and Design: it is a page design and makeup feature that provides WYSIWYG page composition, object and page status tracking, editing, and image cropping directly on the page, page or object templates are used to quickly design pages Table Editing: is a WYSIWYG tool that creates and modifies simple or complex tables; users can import data from external sources such as spreadsheets	Main Application	N
Supervisor		Supervisor provides page, edition and production planning as well as production tracking. It assigns logical pages to physical pages, edition handling, creation of page headers and footers, production of physical pages, typesetting, and imposition operations. It monitors the production process according to deadlines set in Edition Planning. The system controls the entire publication workflow by monitoring the status of pages, objects and editions against predefined deadlines.	Main Application	N






NewsCrap		Image handling, scanning, retouching, cropping and pagination are done with NewsCrap. You can load images from TWAIN interface scanners, directly from the editorial system, from the News Gathering Manager, or from a local or network disk. Within the program, images can be cropped, rotated, mirrored, adjusted, and saved directly to the editorial database, a NewsRoom page, or local or network disk. NewsCrap also lets you browse the News Content Manager database and see thumbnails of all images.	Main Application	N
UPS- Explorer		This application is a container of two editorial systems: News Gathering Manager and News Content Manager. The first editorial system, News Gathering Manager, lets the user store news stories coming from around the world in a database and retrieve them through different queries. The second editorial system, News Content Manager, lets the user search, retrieve and store information in the editorial database. In addition, this tool lets you perform different operations on objects or pages, including unlink, delete, transfer and modify status. You can query objects and pages by name, date created, author, status and other parameters and preview them directly.	Main Application	N
Newsplanner		Used to build the editorial calendar, NewsPlanner lets you schedule coverage of planned events, assign them to reporters and photographers and manage artwork placement. Each department can schedule events on a customizable calendar. Events and other information are stored in a database. From NewsPlanner, you can access NewsRoom or NewsCrap to paginate and release the object or the page to enable just-in-time newspaper pagination.	Main Application	N









ODCenter		The Media Asset Manager is a digital archive that collects and manages data from a variety of sources into one multimedia archive that can be queried from a Windows or Web client interface. Tracking information, such as photo headers or publication data, can be associated with each object, making retrieval a snap. Query tools can be configured to meet the needs of both novices and advanced users.	Main Application	N
ODCConfigurator		Configuration tool used by the administrator to configure and setup all the main functionality of Media Asset Manager.	Utility Application	N
WCAdmin		Configuration tool used by the administrator to manage permissions, groups and public services of News Gathering Manager.	Utility Application	N
NRCConfig		Configuration tool used to define the newspaper environment within the database: sections, headers, footers, zones etc...	Utility Application	N
OMClient		OMClient monitors the printing process, displays the status of all the printer queues, giving to the user the ability to change the priority of a printer job and to start/stop/pause a printer queue.	Utility Application	N
UPSAdmin		Used to administrate users, password, workstation, licenses and permissions for all the Unisys Publication Solution modules.	Utility Application	N
Query		This service is used by News Gathering Manager to display stories already stored in the database.	Query	N









Query + tty		This service is used by News Gathering Manager to display stories already stored in the database (based upon a filter defined by the user) as well as all the new stories coming into the database in real time from wire services.	Query	N
Query + tty with alert		This service is used by News Gathering Manager to display stories already stored into the database (based upon a filter defined by the user) as well as all the new stories coming into the database in real time from wire services. The alert function means that when new story arrives, that meets the user defined criteria, a popup dialog will be displayed showing the new story.	Query	N
Archive query		Archive query is used by News Gathering Manager and Media Asset Manager to query the archive database. In NGM we have two database: the daily db and archive db. In MAM we have just the archive database.	Query	N
Page Query		Page query is used to perform a query of pages in the Hermes database.	Query	N
Object Query		Object query is used to perform a query of objects in the Hermes database.	Query	N
Index to business - Alternative word definition		This service is used to create an index about all the occurrences, inside a newspaper, of a predefined set of words. Option A is used to map a list of alternative words to be used in the generated index to substitute the found strings.	Query	Y









Index to business - Report generation request.		This service is used to create an index about all the occurrences, inside a newspaper, of a predefined set of words. Option R is used to create an index as a text file.	Query	Y
Index to business - Style definition		This service is used to create an index about all the occurrences, inside a newspaper, of a predefined set of words. Option S is used to configure the style of the index to be generated.	Query	Y
Index to business - Index generation report.		This service is used to create an index about all the occurrences, inside a newspaper, of a predefined set of words. Option G is used to create an index as an object into the database.	Query	Y
Referral tool		This service allows the user to search different kind of objects inside the database by name or references.	Query	Y
Get layout		It opens a dialog window used to query in the database all the available layouts that can be linked with the active object.	Query	N
Template Query	NA	Object query is used to perform a query of templates into the Hermes database.	Query	N
Drawer		The drawer is a particular container that can be created by the user in order to fill it manually with the stories chosen by the user. This container can be fed manually by drag and drop of a story from the list directly to the drawer.	Container	N









Drawer autofeeding		The drawer autofeeding is a particular container that can be create by the user in order to fill it manually or automatically by the system with the stories chosen by the user. This container can be fed both manually by drag and drop of a story from the list directly to the drawer and automatically by the system. (Like a TTY)	Container	N
Drawer with alert		This service is designed to be used by groups of users to be filled manually with the stories chosen by the users. The drawer is a particular container that can be created only by the system administrator. This container can be fed only manually by drag and drop of a story from the list directly to the drawer. If a drawer is updated via manual drag and drop, the other users in the group are alerted via a popup window.	Container	N
Drawer autofeeding with alert		This service is designed to be used by groups of users to be filled manually or automatically by the system. The drawer is a particular container that can be created only by the system administrator. This container can be fed both manually by drag and drop of a story from the list directly to the drawer, and automatically by the system based on wire data selection criteria defined by the user. (Like a TTY with alert) . This service is designed to be used by groups of users. If an autofeeding drawer is updated via manual drag and drop, the other users in the group are alerted via a popup window.	Container	N
Repository		The repository is a particular service used only by the News Gathering Manager and is filled manually via drag and drop by the user with stories. This service is very similar to the drawer but more than the previous one this service can be under the security permissions. Only the administrator can create this service and associate it to a particular user or group of users.	Container	N









Repository autofeeding		The repository is a particular service used only by the News Gathering Manager and is filled manually via drag and drop by the user or automatically by the system with stories. This service is very similar to the drawer but more than the previous one this service can be under the security permissions. Only the administrator can create this service and associate it to a particular user or group of users.	Container	N
TTY		This service is used only by the News Gathering Manager to show, in real time, new incoming stories from wire services. The user can create the service associating to it a filter in order to see only a particular kind of agency stories. When a new story is incoming to the system it will be displayed in the application result view.	New agency stories	N
TTY + alert		The tty + alert is a particular service used only by the News Gathering Manager to show, in real time, the new incoming stories from wire services. The user can create the service associating to it a filter to see only a particular kind of agency stories. When a new story is incoming to the system it will be displayed in the application result view and the user is alerted with a popup window containing the text of the new story. (The functionality is similar to the alert associated to the drawer).	New Agency stories	N
Fit to window		This invokes the standard "Fit to window" functionality	System	Y
Colors		It opens the colors window	System	Y



Proof		Invokes the standard "Print" functionality	System	Y
Opened folder		The Open folder is usually displayed in the tree view	System	N
Closed folder		The Closed folder is usually displayed in the tree view	System	N
Window normal		Invokes the standard "window normal" functionality	System	Y
Window cascade		Invokes the standard "window cascade" functionality	System	Y
Window horizontal tile		Invokes the standard "window horizontal tile" functionality	System	Y
Window vertical tile		Invokes the standard "window vertical tile" functionality	System	Y
About		This is the standard "about" button that invokes the "about window".	System	Y

Reload		Invokes the standard explorer like "Reload" functionality	System	Y
Zoom In		Invokes the standard "Zoom in" functionality	System	Y
Zoom out		Invokes the standard "Zoom out" functionality	System	Y
Cut		Invokes the standard "Cut" editing functionality	System	Y
Copy		Invokes the standard "Copy" editing functionality	System	Y
Paste		Invokes the standard "Paste" editing functionality	System	Y
Undo		Invokes the standard "Undo" functionality	System	Y
Redo		It is the standard "Redo" functionality	System	Y





Delete		Delete functionality. Standard functionality	System	Y
Fonts		Opens the font tool window.	System	Y
Up		It is the standard "Up" functionality to scroll up the application results	System	Y
Down		It is the standard "Down" functionality to scroll down the application results	System	Y
Next		Next item in the list item	System	Y
Previous		Previous item in the list item	System	Y
Save		Save an object to local disk	System	Y
Login		Login to the Unisys Publishing System (UPS)	System	Y

Logout		Logout to the UPS system	System	Y
Database		Generic database icon	System	N
Group		Represents a group of users. It's used across all the applications in order to display the user group services	System	N
Single user		Represents the single user. It's used across all the application in order to display the user services	System	N
Scroll up through the selection		Scroll up the result list through the selection	System	Y
Scroll down through the selection		Scroll down the result list through the selection	System	Y
History		Displays the history of the selected object	System	Y
Page		Page is a container of different kind of objects(text, headline, images...) . It is exactly what we see in a newspaper page.	Element	N


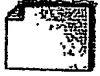




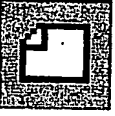
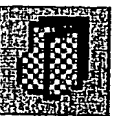
WEB pages		Web pages	Element	N
Mail document		Mail document	Element	N
Acrobat document		Acrobat reader document	Element	N
Sound object		Sound object (It can be a .mid or .avi or any kind of other sound object)	Element	N
Video object		Video object (It can be an avi or mpeg file or any kind of other video files)	Element	N
Image		Image object (It can be a JPG or TIFF or BMP or any kind of other image format)	Element	N
Text		It is the text part of an article, usually made by different columns.	Element	N
Headline		It is the title of the article, usually placed above the text, can be made by different lines.	Element	N








Summary	NA	It is a special block of text that summarizes the article.	Element	N
Caption		It is a short text usually placed above or below an image.	Element	N
Logo	NA	It is a special kind of image, can be an advertisement image, or newspaper sections logo (i.e. "Sport", "Finance" ...)	Element	N
Chart	NA	It is a graph placed on the page.	Element	N
Header	NA	It is a special kind of title, different by the headline, it usually identifies the newspaper sections (i.e. "Sport", "Finance" ...).	Element	N
Table		This is a table object	Element	N
Classified Ad	NA	This is a classified advertisement. Examples: job research, sell & buy...	Element	N
Obituary	NA	This is an advertisement for a death.	Element	N









Ad Space	NA	This is a "zone" on the page reserved for advertisement image.	Element	N
Ad	NA	This is a "zone" on the page reserved for advertisement text.	Element	N
Ad Rules	NA	Special kind of boundary lines "rules", reserved for the advertisement objects. (Also see Rule and Freehand Rule - each are similar in concept)	Element	N
Box	NA	This is a box that can be drawn on the page around a text or a headline (also filled inside with different colors).	Element	N
Rule	NA	This is a horizontal, vertical or diagonal line "rule" that can be solid, dashed or dotted. These special kind of lines "rules", are placed on the left, right, top or bottom of a text object.	Element	N
Polygon	NA	This is a geometric object, having the desired number of sides.	Element	Y
Ellipse	NA	This is a geometric object, having the desired size and thickness.	Element	Y
Box with rounded angles	NA	Exactly as the object type "Box", but with rounded angles.	Element	Y

RunAround	NA	This is a special effect: text that flows around an image or a contour drawn with a graphical object (ellipse, box...).	Element	Y
Freshhand rule	NA	This is a free hand boundry line "rule" made up of two or more sections ("rule lines").	Element	Y
Header		Header of the page	Element	Y
Footer		Footer of the page	Element	Y
Graphycal page		Type of page.	Element	Y
Section	NA	To associate a page to a configured section (NEWS, SPORT, FINANCE...) within Supervisor.	Element	Y
Signature	NA	Signatures facilitate the correct page association with plates and ordering of the physical pages, to ensure the paper will be printed in the correct sequence.	Element	Y
Newspaper		The entire newspaper	Element	Y









00157277 002502










Subject symbol		The subject symbols are symbols that are used to quickly find a subject among the large mass of information of a newspaper. Symbols are organized according to a hierarchical tree that envisages a Parent/Child Peer to Peer relationship.	Element	N
New Page		Opens a dialog window to insert all the data to create a new page into the database (similar to the "New" functionality of Word)	Element actions	Y
Open Page		Opens a dialog window to insert all the data to open an existing page from the database (similar to the "Open" functionality of Word)	Element actions	Y
Save Page		To save the active page into the database	Element actions	Y
Release Page		This action changes the status of a page in the system.	Element actions	Y
Quit Page		To close without saving the open page	Element actions	Y
New Object		Opens a dialog window to insert all the data to create a new object (text, headline, table etc...) in the database (similar to the "New" functionality of Word)	Element actions	Y
Open Object		Opens a dialog window to insert all the data to open an existing object (text, headline, table etc...) from the database (similar to the "Open" functionality of Word)	Element actions	Y










Save Object		To save the active object in the database	Element actions	Y
Release Object		This action changes the status of an object in the system.	Element actions	Y
Quit Object		To close without saving the active object	Element actions	Y
Typeset page	NA	The "typeset" is a particular kind of print functionality, different by the normal print on a local printer (i.e. a common laser printer), it is the final printing process, in this case the page is sent to the newspaper printing press.	Element actions	Y
Typeset object	NA	Same as "Typeset page"	Element actions	Y
Delete header		Header of page deleted	Element actions	Y
Delete footer		Footer of the page deleted	Element actions	Y
Stand alone		Opens a stand alone window that contains the selected object.	Element actions	Y
Lock edition		This functionality allows the user to lock the edition in order to disable any other user modifications.	Element actions	Y









Engagement		Engagement function allows the user or the system to engage an object in order to store additional information that advises all other users that the object has already been used by another user.	Element actions	Y
Lock		Generic lock action for any element	Element actions	Y
Unlock		Generic unlock action for any element	Element actions	Y
Lock physical page		This functionality allows the user to lock the physical page in order to disable any other user modifications.	Element actions	Y
Lock newspaper		This functionality allows the user to lock the entire newspaper in order to disable any other user modifications..	Element actions	Y
Print product		This function allows the print preview of pages associated to the plates. (Peter; this is the final print just prior to the typeset step. Let me know if this is clear.)	Element actions	Y
Textual print		Print of only text objects on a page.	Element actions	Y
Makeup window		To open the Makeup window, a tool to search pages and objects into the database	Tool windows	Y
Templates window	NA	To open the Templates window, a tool to search page or object templates	Tool windows	Y









18 - February - 2003 New Icons









Graphical print		Prints the output in graphical format Peter - This is the second half of a pair. The other half of the pair is Textural printer previously provided above in row 123.	Element actions	Y
IPTC Info		Standard IPTC information related to an image and usually stored inside the image	Image action	Y
Zoom 1:1		Classic functionality of ZOOM 1:1 in order to display the image in the standard size	Image action	Y
Mirror vertical		Mirror vertical of an image	Image action	Y
Mirror horizontal		Mirror horizontal of the image	Image action	Y
Curves		Draw curve line within an image	Image action	Y
Rotate		Rotate image	Image action	Y
Rotate 90° to left		Rotate image 90 degree to left	Image action	Y









Rotate 90' to right		Rotate image 90 degree to right	Image action	Y
Invert colors		Invert colors of the image	Image action	Y
Crop parameters		Set crop parameters. This function allows the user to set the crop for the image	Image action	Y
Get crop		Get crop allows the user to get the crop from another running application. Example: Using news gathering Manager with an image opened, it is possible to get the crop information from newsroom layout while it is running.	Image action	Y
Toggle quality high		After a BINUSCAN filter of the image, this icon represents the end status of image quality. In this case the icon represents high quality. Clicking this button, the image is converted to low quality	Image action	Y
Toggle quality med		After a BINUSCAN filter of the image, this icon represents the end status of image quality. In this case the icon represents medium quality. Clicking this button, the image is converted to low quality	Image action	Y
Toggle quality low		After a BINUSCAN filter of the image, this icon represents the end status of image quality. In this case the icon represents low quality. Clicking this button performs no action.	Image action	Y
Open local image		Open local image from the hard disk	Image action	Y
Binuscan filter		Apply the BINUSCAN filter	Image action	Y



Scan slide		Scan slide with predefined slide scan settings	Image action	Y
Select source		Select input source of the image	Image action	Y
Scan paper image		Scan image with predefined image scan settings	Image action	Y
Clipping path options		This action allows the user to set the clipping path parameters inserted by photoshop. (It's a photoshop function) The user can decide to see the entire photo or the clipped portion of the photo.	Image action	Y
Media Asset Manager properties		Generic properties icon	Settings	Y
Enable web link		Enable web link inside of the text. If a web link is found inside of the text it will be considered as an active link.	Settings	Y
Disable web link		Disable web link inside of the text. If a web link is found inside of the text it will be considered like normal text	Settings	Y
Book template		Allows the user to create a preview of the newspaper	Settings	Y
Grid mode		Change the view from list (slug) to grid (thumbnail) mode	Settings	Y

List mode		Change the view from grid (thumbnail) to list (slug) mode	Settings	Y
Restore panes		This function allows the user to restore the original dimensions of the three application frames	Settings	Y
Squeeze text		This function lets the user display the text in the stand alone window or object frame in "fit to window" modality or "word wrap" modality. It's used when a text line is longer than the window frame size.	Settings	Y
Dropcap		To apply the dropcap effect to the active text; dropcap is the big initial caps letter used to start a story	Text editing	Y
Lowercase		Sets selected text to lower case	Text editing	Y
Uppercase		Sets selected text to upper case	Text editing	Y
Statistics		Used to display statistics (number of words, characters, lines...) about the active text.	Text editing	Y
Thesaurus		Used to open the thesaurus window while editing text	Text editing	Y

Spelling checker		Used to spell check the active text, to find spelling and syntax errors	Text editing	Y
Savegets		It opens a window to configure accelerator keys to speed up the insertion of predefined strings into the active text	Text editing	Y
Macro		Opens a window to select a previously recorded macro and run it on the active text. A macro is a sequence of text editing actions.	Text editing	Y
Change language		Used to change the predefined language used to edit texts (spell checker, hyphenation...)	Text editing	Y
Import local text		Used to open a window to browse the local harddisk in order to select a text file to be imported into the active text.	Text editing	Y
Find		Used to open the Find & Replace dialog window to be used on the active text.	Text editing	Y
Signature tag		Used to insert the user signature at the end of the text in editing	Text editing	Y
Text tag		Used to insert the special id (=tag) text in the active text	Text editing	Y

Headline tag		Used to insert the special id (=tag) for headline in the active text	Text editing	Y
Define		This icon invokes the "lasso" functionality, to select several objects simultaneously in the page	Layout handling	Y
Draw layout		When this functionality is active, the user can draw box containers in the page to be filled with text, images, etc...	Layout handling	Y
Merge objects		Used to group several previously selected layouts making them a unique layout. Layouts are selected using the "Define" functionality.	Layout handling	Y
Full page		Used to invoke a predefined zoom factor to get a view of the entire active page	Layout handling	Y
Half page		Used to invoke a predefined zoom factor to get a view of half of the entire active page	Layout handling	Y
Delete row		To delete the selected row of a table.	Layout handling	Y
Delete column		To delete the selected column of a table.	Layout handling	Y

Insert row		To insert a new row into a table.	Layout handling	Y
Insert column		To insert a new column into a table.	Layout handling	Y
Join vertical		To merge selected vertical cells within a table. (convert multiple vertical cells to one)	Layout handling	Y
Join horizontal		To merge selected horizontal cells within a table. (convert multiple horizontal cells to one)	Layout handling	Y
Split vertical		To restore the selected vertical cell, that was previously merged, back to the original cell configuration.	Layout handling	Y
Split horizontal		To restore the selected horizontal cell, that was previously merged, back to the original cell configuration.	Layout handling	Y
Row x column		To modify the number of rows and column of the selected table.	Layout handling	Y
Rule - All		This icon is placed on a push button used to apply a previously selected style to all the borders of an object layout.	Layout handling	N

Assignment		A process to define the jobs to be done by single users and/or teams	Event management	Y
Event		A topic of editorial interest that will be assigned to a specific user or team.	Event management	Y

60457237 -032501



Third Parties Integration with NewsRoom Project Functional Specification

HE70-FS-TPIN.doc

**Authorized for issue
by:**

U Gadarco

**Unisys
Project Manager**

Signature

Date

H Hofmann

**Unisys
Editorial Product
Manager**

Signature

Date

**December 19, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Third Parties Integration with NewsRoom Project Functional Specification				
Author	BC Requirements Analyst Unisys				
To Be Approved By	UG Project Manager Unisys		HH Editorial Product Manager Unisys		
Comments	This is the Functional Specification for the Third Parties Integration with NewsRoom project				
File Name	HE70-FS-TPIN.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	12/19/2002	BC	First version of the Functional Specification

Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	5
2.3	ASSUMPTIONS AND DEPENDENCIES	6
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE	6
3	FUNCTIONAL DESCRIPTION.....	7
3.1	HANDLING EXTERNAL PAGES IN NEWSROOM	7
3.1.1	Description.....	7
4	INFORMATION FOR USER MANUAL	10
5	PERFORMANCE	10
5.1	SPEED.....	10
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY	10
5.3	CAPACITY	10
6	INSTALLATION	10

1 References

This functional specification refers to Third Parties Integration with NewsRoom Editorial Enhancement.

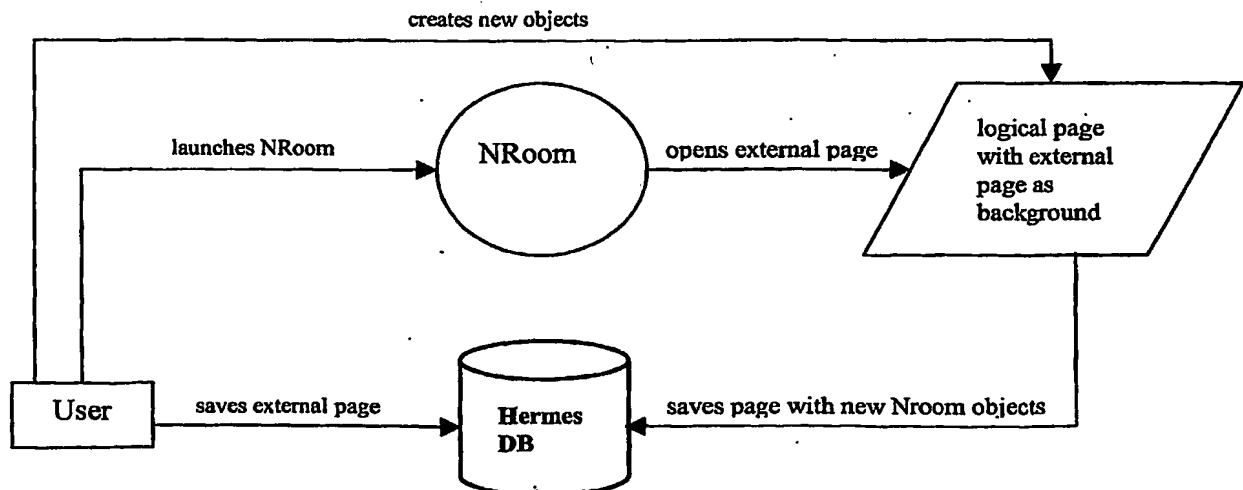
2 Functional Overview

The integration of third parties applications with the Hermes applications will imply some changes in the Hermes NewsRoom module, in order to handle pages coming from external applications.

2.1 Major functions

NewsRoom will be modified in order to be capable to read and edit pages created with third parties application and specifically with the Adobe InDesign application.

2.2 Data/Activity Diagram



2.3 Assumptions and Dependencies

- The figures depicted in this document are dummy screenshots.
- The GUI of all new dialog windows/controls implemented must be XP-like style.
- Previous NewsRoom behavior to handle pages coming from Hermes will be maintained

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.

3 Functional Description

The integration of third parties applications with the Hermes editorial systems requires the Hermes modules to be partially modified in order to interact with external applications.

This project aims at modifying the NewsRoom behavior so that this module is capable to handle pages coming from external applications and, specifically, from Adobe InDesign application.

3.1 Handling external pages in NewsRoom

3.1.1 Description

Pages coming from external applications will be imported in NewsRoom and handled as images that will be used as a background of the NewsRoom logical page. External pages will be treated as image type objects and users will not be able to modify them. The image displayed as background will be generated using a preview of the external page.

Once an external page is imported and displayed as background of a Newsroom page, users will be able to take actions such as creating new objects, saving, etc., as currently occurs when working on a standard NewsRoom page.

If the page is open in Layout mode the page grid will overlap the external page. The overlapping of the grid will be transparent, thus allowing users to see the objects making part of the external page. In this way, users will be able to create new objects in NewsRoom by taking into consideration the already existing objects while designing new elements.

The NewsRoom page will be composed of different layers: the external page (along with its objects) in the background, the page grid (if displayed) in the middle, and the new NewsRoom objects (if any) displayed in the foreground. This means that it will not be possible to create layout that automatically clip the external page layouts. Newsroom elements added to the page will overlap the EPS representing the preview of the external page.

XMP technology will be used for embedding metadata into InDesign and InCopy objects.

Figure 1 – Example of Newsroom page open in Layout mode with background external page

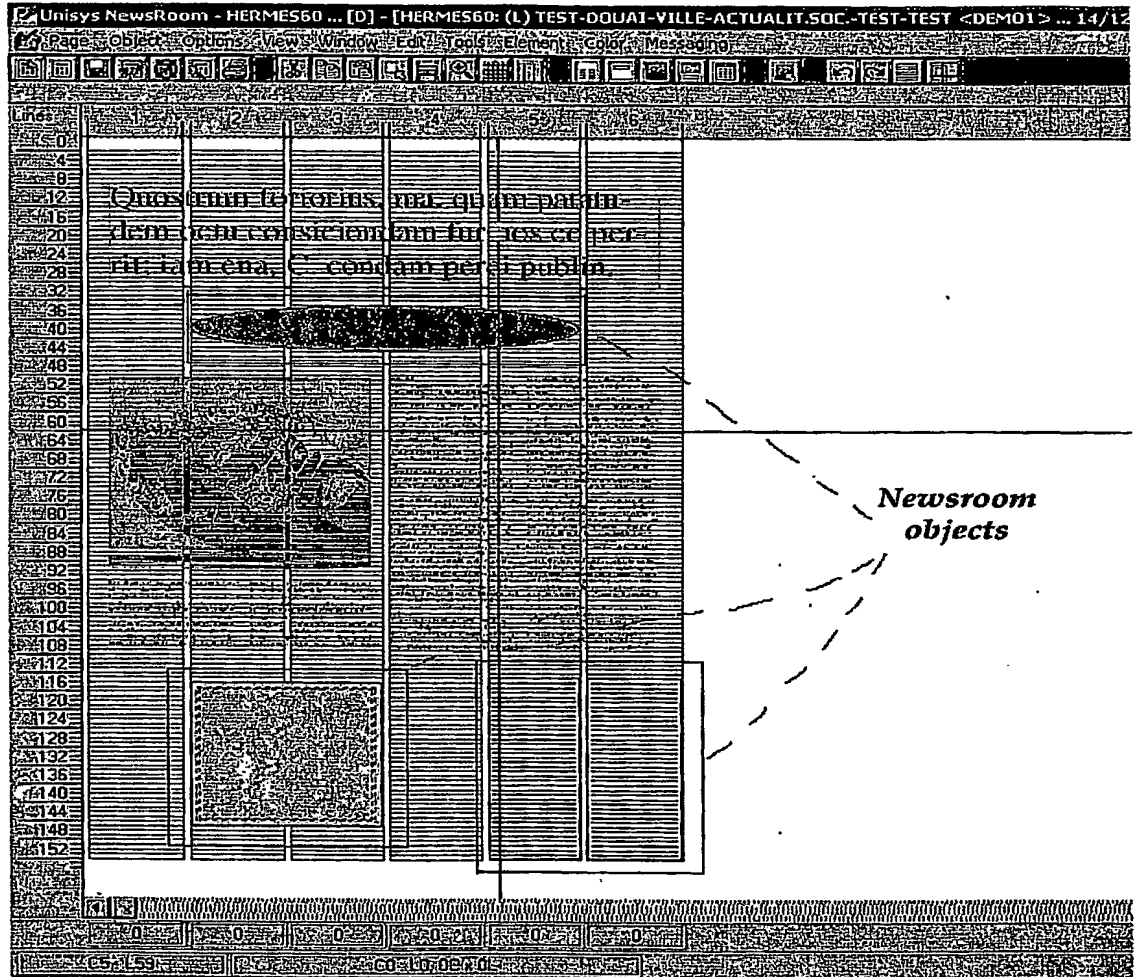


Figure 2 – Example of Newsroom page open in preview mode with background external page



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.



Workflow Definition for External Objects Project Functional Specification

HE70-FS-WDEO.doc

**Authorized for issue
by:**

A Politi

**Unisys
Adobe Integration
and Integration
Platform Project
Manager**

Signature

Date

H Hofmann

**Unisys
Editorial Product
Manager**

Signature

Date

**January 28, 2003
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Workflow Definition for External Objects Project Functional Specification				
Author	BC Requirements Analyst Unisys				
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys			HH Editorial Product Manager Unisys	
Comments	This is the Functional Specification for the Workflow Definition for External Objects project				
File Name	HE70-FS-WDEO.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	01/28/2003	BC	First version of the Functional Specification

Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	5
2.3	ASSUMPTIONS AND DEPENDENCIES	5
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE.....	6
3	FUNCTIONAL DESCRIPTION.....	7
3.1	DEFINING A WORKFLOW FOR EXTERNAL OBJECTS	7
3.1.1	Description.....	7
4	INFORMATION FOR USER MANUAL	9
5	PERFORMANCE	9
5.1	SPEED.....	9
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY.....	9
5.3	CAPACITY	9
6	INSTALLATION	9

1 References

This functional specification refers to Workflow Definition for External Objects Editorial Enhancement EE-2002-03.

For the description of the functions available when defining Hermes pages and objects workflow using the NrConfig module, refer to the NrConfig User Manual (HEXX-DOC-NRCUM)

2 Functional Overview

The integration of third parties applications with the Hermes applications will imply some changes in the Hermes NrConfig module, in order to allow handling objects coming from external applications, so that a specific workflow can be appropriately defined for them.

2.1 Major functions

Capability to define a specific workflow for objects created with third parties applications, so that they can enter and follow all the production phases as currently occurs for standard objects and pages created with Hermes client applications.

2.2 Data/Activity Diagram

Not applicable.

2.3 Assumptions and Dependencies

- The figures depicted in this document are dummy screenshots.
- The GUI of all new dialog windows/controls implemented must be XP-like style.
- The current functions used to define the workflow for Hermes objects/pages will not vary. The workflow of external objects will be defined using the same functions available for Hermes objects.

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.

The integration of third parties applications with the Hermes editorial systems requires the Hermes modules to be partially modified in order to interact with external applications.

3.1 Defining a workflow for external objects

As already available for Hermes objects and pages, users will access the Nrconfig module to define a specific workflow also for objects coming from third party applications.

In order to handle external objects, a new tab will be added to this dialog window, as depicted in the figure below.

The screenshot shows the Copyflow designer window with a state transition diagram. The diagram consists of the following states and transitions:

- EXTINT** (Start State)
 - Transition to **EXTREADY** labeled **EXTREADY -> EXTINT**
 - Transition to **EXTINTD** labeled **EXTMD -> EXTINT**
- EXTREADY**
 - Transition to **EXTINT** labeled **EXTREADY -> EXTINT**
 - Transition to **EXTMD** labeled **EXTMD -> EXTREADY**
- EXTINTD**
 - Transition to **EXTMD** labeled **EXTINT -> EXTMD**
- EXTMD**
 - Transition to **EXTINT** labeled **EXTMD -> EXTINT**
 - Transition to **EXTREADY** labeled **EXTMD -> EXTREADY**
 - Transition to **EXTFINAL** labeled **EXTMD -> EXTFINAL**
- EXTFINAL** (End State)
 - Transition to **EXTMD** labeled **EXTFINAL -> EXTMD**

Workflow Definition for External Objects

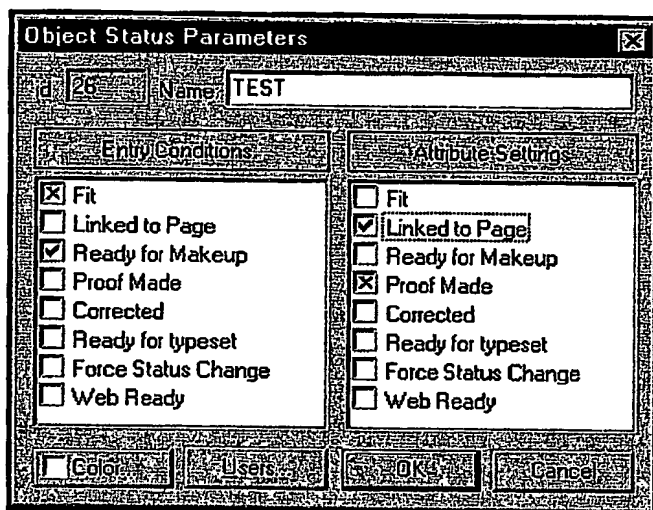
Unisys Proprietary

Users will use this tab in the same way the relevant tabs are used for Hermes objects and pages.

It will be possible to define statuses for external objects by selecting the *New Status* radio button and drawing the circles corresponding to the status being defined.

Users will then be able to define the entry conditions and the attributes for each of the designed statuses by enabling the *Select* radio button that will display the **Object Status Parameters** dialog window.

Figure 2 – The Object Status Parameters dialog window



By using this dialog window, users will also be able to associate a color with each defined status by clicking the relevant button and selecting a color from the displayed dialog window.

In addition, it will be possible to specify the user the object released in the defined status must be sent to, through the list of users displayed in a separate window when clicking the *Users* button.

Of course, the External Objects tab will also allow to define transitions between the statuses the object will move to (*Transition* radio button) and delete statuses (*Delete* button).

In addition, it will be possible to copy the workflow defined for Hermes objects and use it for external objects.



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.



Adobe InDesign *Integration with News Content Manager – Hermes*

Internal Use Only

Copyright © 1999 Unisys Corporation.
Unisys is a Unisys Corporation Trademark.
Hermes™ is a trademark licensed to Unisys Corporation

Milan, May 2002

UPS-MKT-STG

Rev. 00

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. Information contained in this document is highly sensitive, confidential and of a competitive nature. Any non-Unisys party participating in the preparation or review of this document shall respect the confidential nature of the information contained therein.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT.

Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Unisys is a registered trademark of Unisys Corporation

Hermes is a trademark licensed to Unisys Corporation

**Copyright © 1997 Unisys Corporation
All rights reserved**

Document Information

Document Title	Adobe InDesign Integration with News Content Manager – Hermes				
Author	Massimo Barsotti - Unisys				
To Be Approved By	Project Manager - Unisys				
Comments					
File Name	Adobe InDesign Integration.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Author	Remarks
1.0				M. Barsotti	
1.1	2.4		12062002	H. Hofmann	
1.2	4		12062002	M. Barsotti	
1.3	5		13062002	M. Barsotti	
1.4	All		13062002	H. Hofmann	
1.5	All		14062002	H. Hofmann	
1.6	1.2		14062002	M. Barsotti	

TABLE OF CONTENTS

* * *

1. Introduction.....	5
1.1. <i>Benefits.....</i>	5
1.2. <i>Limitations.....</i>	5
2. System architecture	6
2.1. <i>Integration overview.....</i>	6
2.1.1. The following platform-related scenarios will then be available:.....	7
2.1.2. Copy-driven workflow.....	7
2.1.3. Layout driven workflow	8
2.2. <i>Architecture overview.....</i>	9
2.3. <i>Integration components</i>	9
2.4. <i>New developments</i>	10
2.4.1. Hermes palette tool	10
2.4.2. InCopy/InDesign plugins.....	12
2.4.3. Hermes Web Agent.....	13
3. Database related topics.....	13
3.1. <i>Pages.....</i>	13
3.1.1. Database requirements	13
3.2. <i>Texts.....</i>	14
3.2.1. Database requirements	14
3.3. <i>Importing NewsRoom texts</i>	14
3.3.1. Preserving text features.....	15
3.4. <i>Importing graphical content.....</i>	16
3.5. <i>Photoshop images.....</i>	16
3.6. <i>Illustrator objects, PDF files.....</i>	16
4. Hermes configuration	17
5. What's left.....	17

1. Introduction

This document describes a possible scenario for the integration of Adobe applications with the Hermes editorial system, considering InDesign as the pagination tool and InCopy as the text editor.

Main goal of this document is to describe the technical architecture, layout the required functionality, highlight potential problems and limitations and provide a basis for a detailed technical evaluation/design to be performed by the development team(s) (internal and external).

1.1. Benefits

This approach will merge the high-end graphic functionalities of InDesign with the production capabilities of the NCM-Hermes Publishing system, and Adobe applications will become a viable alternative to Unisys tools.

Major benefits:

- Users will have the possibility of choosing InDesign or NewsRoom to create pages;
- Integration will be available on both Windows and Mac platform;
- No more conflicts between features compatible and not compatible with NewsRoom;
- Editorial content will be accessible from InDesign and InCopy;
- Supervisor will remain as the main production tool for managing editions and output;
- NewsPlanner will remain available as the news-planning tool;
- Hermes workflow will remain and will be available in InDesign and InCopy;
- The solution with NewsRoom entirely replaced with Adobe tools will be appealing to magazines.

1.2. Limitations

Some limitations of the integration (that is, Hermes features that will not be available in InDesign) are listed below: the list should not be intended as definitive, given that viable alternatives may be found using InDesign features or by incorporating third party plugins (see <http://idplugins.com/catalog/>, <http://thepluginsite.com>).

- Inheritance of pages and objects between editions
- Jump stories
- Automatic fields (e.g. jump lines, folios)
- Vertical copyfit
- Pagination tools (e.g. grow, snap, move, etc.)
- Messaging
- Autolink
- Indexing tools
- Language support (*)

(*) While InDesign is currently available in all the most important languages, (different versions will ship at different times) InCopy is currently available only in English, French, German, Italian, Spanish, and Swedish: Adobe mentioned the possibility of adding language support where strongly required.

Besides these, the integration as described below, is only focusing on how these Adobe tools can be integrated into the overall UNISYS publishing solution. It does not cover any client related function, that is possibly required to make InCopy and especially InDesign a more productive tool and more applicable in the newspaper environment.

2. System architecture

2.1. Integration overview

This integration follows a different approach from what is available in Release 6.0: it is more focused on making InDesign and InCopy part of the suite of Hermes tools, allowing them to access content stored in the database, save pages and stories in the database, support workflows, user permissions and profiles.

It is less focused in making InDesign and NewsRoom closer or more "compatible". An effort has to be done to avoid replicating features in NewsRoom: in fact, the two applications will remain and will coexist with a limited interaction.

Usage of a "neutral" layer to communicate with Hermes will be kept: the application will use an XML-based protocol to manage metadata information back and forth from the database.

Basically, the idea is to give InDesign the capability of creating "logical pages" in Hermes: these will be distinct from pages created with NewsRoom, and will be managed by InDesign only. Using InDesign, pages can be created with the contribution of other Adobe tools (e.g. Photoshop, Illustrator), with InCopy playing a main role as the text editor: these pages will be managed using Supervisor (eventually will be merged with other logical pages) and will be part of the production workflow.

A very important aspect of this integration is, that we will not loose our main concept in Hermes of being able to modify pages and object at the same time by different people. As this is not a standard functionality of InDesign and InCopy, it needs to be implemented as part of the plugin. WoodWing (our development partner) indicated, that they have such a plugin (based on file system) available, which we have evaluated and it seems to be the right starting point for this integration.

We will then have a new separate production process with a second set of tools, merging with the traditional one in the Supervisor. In this scenario, pages (and objects) can only be modified with the tool that was used for the page (object) creation. Moreover, exchange of textual content between Hermes and Adobe tools will be based on "copies" of the content itself: a conversion will be required (with a risk for limitations) and workflow features (lock/unlock, statuses and permissions) have to be carefully maintained.

2.1.1. The following platform-related scenarios will then be available:

Windows-based scenario:

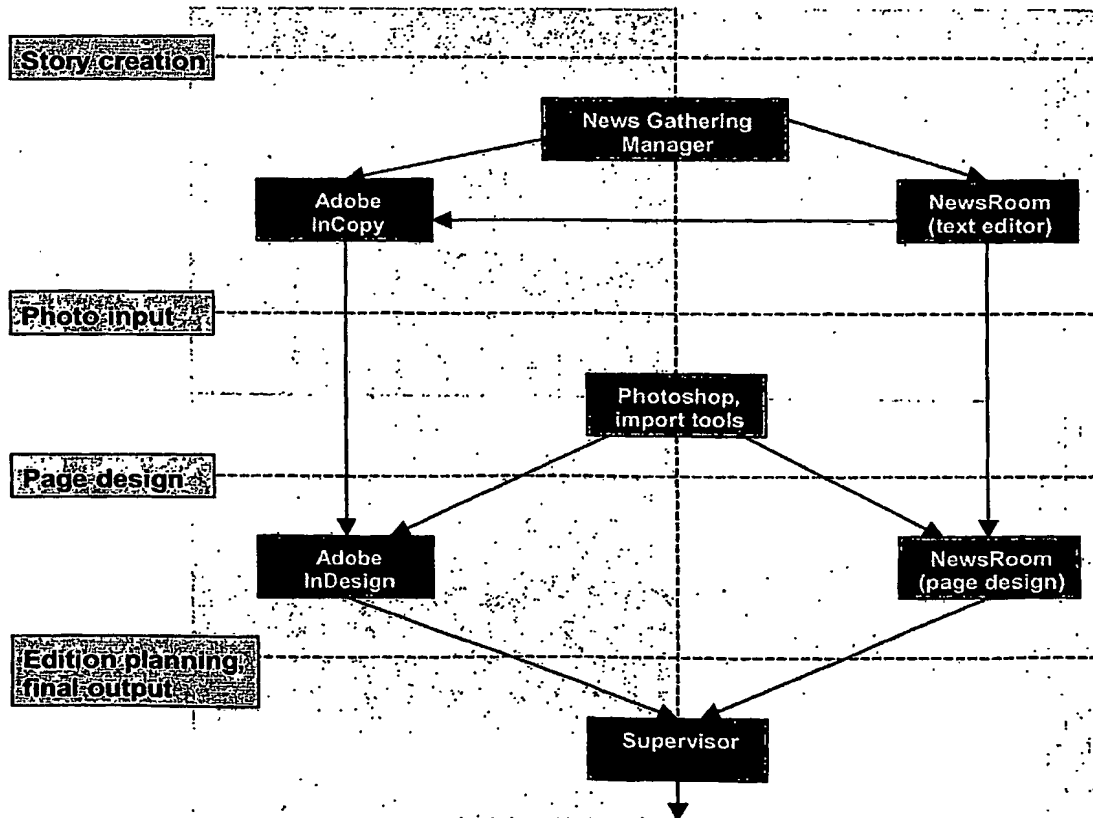
Department	Applications
Editorial	InCopy, NewsRoom, UPS-Explorer (NGM)
Photo	Photoshop
Layout	InDesign, NewsRoom
Production	Supervisor, UPS-Explorer

Mac-based scenario:

Department	Applications
Editorial	InCopy, NGM-Web
Photo	Photoshop
Layout	InDesign
Production	Supervisor, UPS-Explorer (<i>Windows</i>)

2.1.2. Copy-driven workflow

The diagram below shows, how this integration supports copy-driven workflow scenarios.



Story creation: texts can be created using either NewsRoom or InCopy, and saved in Hermes database. In both cases, News Gathering Manager can be used to access wires or contributions from reporters (on the Mac, the browser-based client can be used). It will be possible to import texts created with NewsRoom in InCopy.

Photo input: all the tools available today will remain available and will be used to import images and graphic files in the Hermes system (database and filesystem).

Page design: pages (Hermes logical pages) can be created using either NewsRoom or InDesign, and saved in Hermes database. InCopy texts can be paginated in InDesign without composition problems, due to the integration between the two applications. Images stored in the Hermes database will be paginated, using the low-res version, and proper OPI links to high-res files will be inserted.

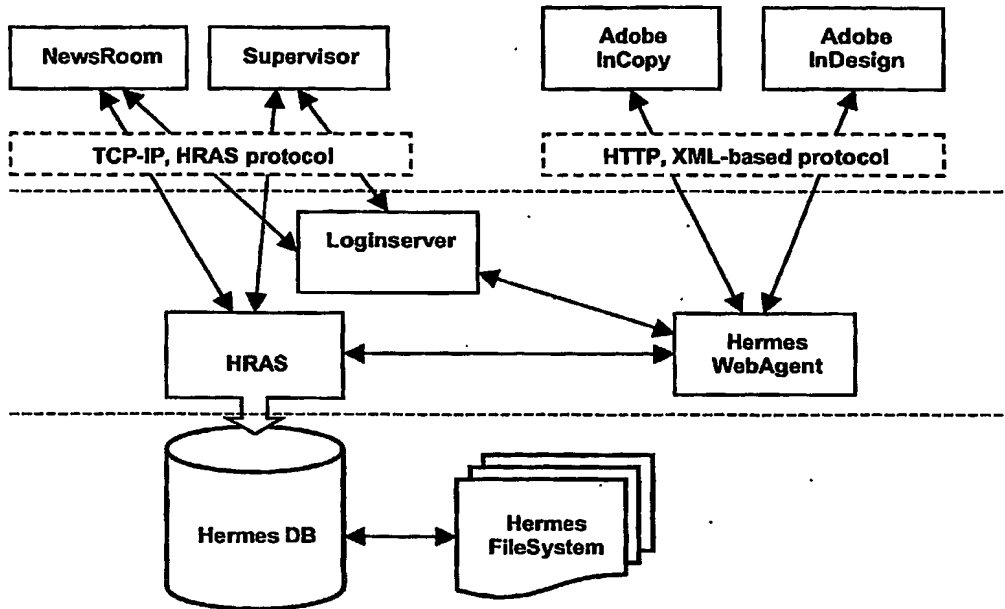
Edition planning, final output: Supervisor will be used, as today, to manage publications, editions and to drive final output of all pages: from the Supervisor, it will be possible to mix logical pages created with InDesign and NewsRoom, and to track the production of all the pages in the system. Supervisor will remain available on Windows only, while the tracking functionalities will be accessible from the Mac using the browser-based client.

2.1.3. Layout driven workflow

Fully supported by the integration. It will be possible to design a page first, then assign the various text containers (also complex containers with several elements) to different InCopy objects: editors assigned to write a story will open the text object only, with the layout designed in page, and will write to fit. InDesign and InCopy use the same composition engine, therefore hyphenation, line endings and all the typographical aspects will be maintained.

2.2. Architecture overview

Various integration components will communicate with the server (and the database) in different ways, summarized in the diagram below:



While Hermes applications, like NewsRoom and Supervisor, will continue to communicate directly with server applications (typically HRAS and Loginserver), relying on TCP-IP and HRAS protocol, Adobe applications will follow a different path.

InCopy and InDesign will not “talk” directly with server processes, but will communicate with Hermes Web Agent, a new server module introduced in Release 6.0 to manage communication with Web clients. Hermes Web Agent, in turn, will be connected with HRAS and LoginServer to accomplish user validation and database access.

InCopy and InDesign will rely on the XML protocol, HTTP-based, already introduced in Release 6.0: it is platform independent, ensuring that the integration will be available on Macs, and successfully used in current implementation.

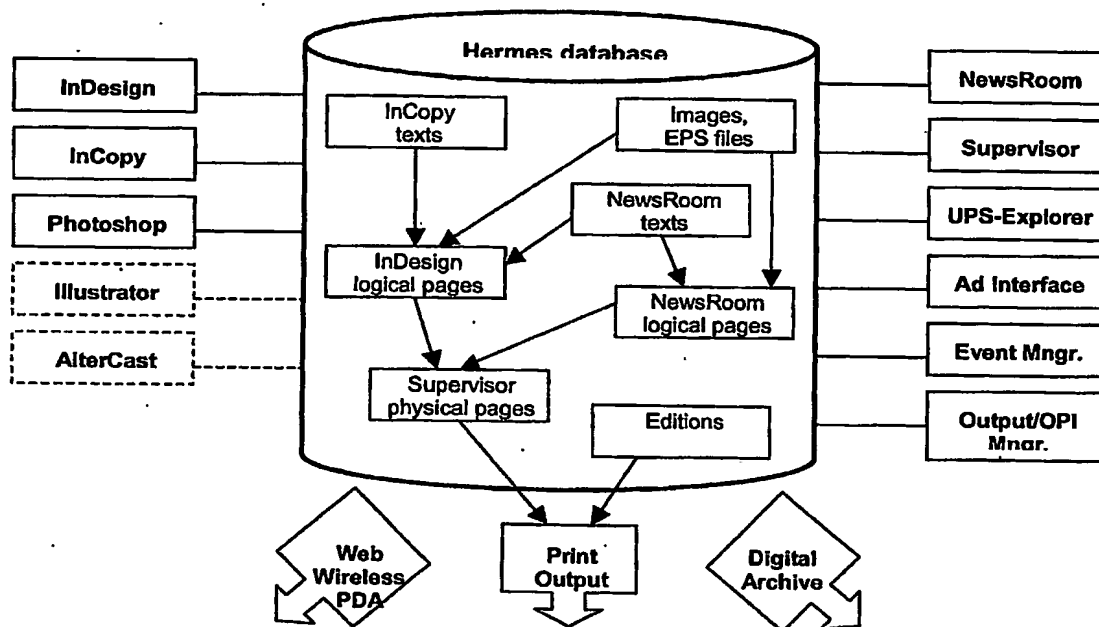
2.3. Integration components

The diagram below collects all the actors of the integration and the objects managed by the applications.

Hermes database plays a central role as the common repository for content and metadata used both by Hermes client applications (e.g. NewsRoom, Supervisor, etc.) and by Adobe programs (InDesign, InCopy). Applications interested by the integration are on the two sides of the diagram: Photoshop is already integrated with Hermes. Illustrator and AlterCast, even if mentioned in this document because of their potential roles, are not included in this phase.

The database stores:

- Graphic files, either imported or saved using Photoshop, to be paginated either with InDesign or NewsRoom;
- NewsRoom texts, to be paginated with NewsRoom or to be imported in InCopy and paginated with InDesign;
- InCopy texts, to be paginated with InDesign;
- NewsRoom logical pages;
- InDesign logical pages;
- Physical pages created from logical pages, even from merging NewsRoom and InDesign logical pages;
- Edition information.



Supervisor manages output for print editions, while content can be extracted for archiving or re-purposing in digital format.

2.4. New developments

The architecture introduced above includes standard and new components: the following is a list of the most significant new developments required by this architecture.

2.4.1. Hermes palette tool

This application will be available in both InDesign and InCopy as a palette and will provide access to Hermes database, respecting workflow and user permissions. It will resemble NewsRoom "makeup window", listing object and pages, displaying previews, opening items via drag & drop, etc. It will communicate with Hermes Web Agent to access Hermes.

The following list contains the functionalities required:

- Search: The palette should provide an simple fast search function capable of defining searches on all important fields like
 - Object/Page name
 - Level
 - Publication date
 - Author
 - Status
 - Type (INDD, INCD first, then image/graphic objects, last NR textual objects)

In addition to this fast search, a more complex/extended search should be possible providing the same search functionality as the UPS Explorer currently provides. In addition to the fields mentioned above, the following fields should be accessible:

- Object/Page comment
- Edition
- Creation, Modification, Modifying users and intervalls
- Workflow fields (used, locked, in page, outside page, deleted)

For InDesign page, some additional search options should be implemented:

- Pages plus all attached objects (INCD and image/graphic objects)

The palette itself should support different sorting options, preferably by clicking into the table header field. The fields available for display should be configurable.

- Link: The palette is the main tool to paginate objects onto InDesign pages. This tool should implement the following functions.
 - Link INCD elements to a INDD page: This function will place the INCD document onto the INDD page by establishing a link (like in WoodWing smart connections). The UPS database needs to be updated immediately after this action, so the INCD object can be marked as "Paginated".
 - Link NR image/graphic elements to a INDD page: This function will place the image/graphic object as it is. The UPS database needs to be updated immediately after this action, so the object can be marked as "Paginated". The low-res image has to be passed to INDD. During output, the low-res path has to be converted into the hires path before the OPIFILTER is running and processing the file.
- Unlink object: The object needs to be marked as "Available" again in the UPS database.
- Open INCD object or INDD page: By using drag&drop onto the desktop, either the object or the page should be opened in the application. The UPS database needs to be updated immediately after this action, so the object/page can be marked as "Locked".
- Delete INCD object or INDD page: The UPS database needs to be updated immediately after this action, so the object/page can be marked as "Deleted".

- Copy content from NR text object into a INCD object: This function should import the content of the selected NR textual object and import it into the selected story in INCD or INDD. Conversion of typographical commands and tags need to be implemented (to be defined). The source object should get a marker either in the comment field or using particular status indicating, that this object is in being used in the INCD/INDD world.

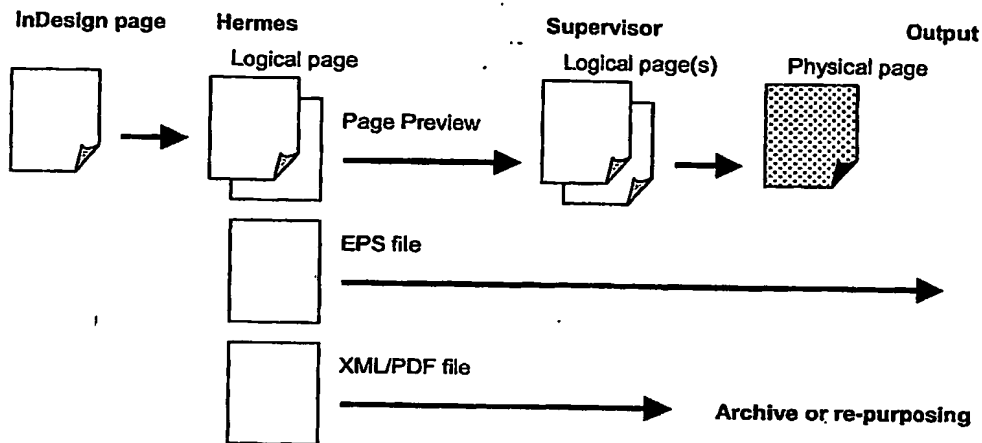
2.4.2. InCopy/InDesign plugins

The interaction between Adobe application and Hermes palette will be managed by a new set of plugins specifically developed: they will implement the proper behaviour in both applications.

In addition to the above-mentioned functions, the following list of functionalities needs to be provided within InDesign or InCopy.

- Create new INDD page: Like in the existing plugin, a new page should be generated based on available page grids. This function should immediately
 - create a new record for a logical page in the UPS database and lock this record
 - create a INDD/graphic object and place this on to the logical page and lock it in the UPS database.
 - set initial status: this will be part of a specific workflow that will be available for InDesign/InCopy objects. Hermes configuration tool will be updated to allow the definition of the new workflow.
- Save INDD page: The record of the INDD object needs to be updated in the database and the INDD doc need to be transferred to the UPS server.
- Release INDD page: This function should
 - Update the logical page record accordingly (status etc) and unlock the logical page.
 - Update the object record for the INDD/graphic object and unlock the object record.
 - Depending on the status, create an EPS and lock all INCD object records in the UPS database (meaning, send them into a no-editable status)
- Create new INCD document: Create a new object record in the UPS database and lock this record.
- Save INCD document: The record of the INCD object need to be updated in the database and the INCD doc need to be transferred to the UPS server.
- Release INCD document: This function should update the INCD object record accordingly and unlock the object record.
- Move/copy INDD or INCD object:
- Unlock INCD or INDD object:
- Change status of INCD or INDD object:

To show with a simple example, let's assume a single page has been created using InDesign and has been save in the database. From this point on:



Multiple logical pages: When the multi-page document is completed, it needs to be saved in Hermes as several logical pages. The user will be prompted for as many logical pages as pages in the InDesign document: for each logical page, an EPS and a preview will be generated (the multi-page EPS will be split in several EPS files, this is supported by InDesign). XML and PDF versions will be generated as well.

2.4.3. Hermes Web Agent

This application will be enhanced – where needed – in order to fulfil all the need of the integration. New functionalities may be added (e.g. create new page) and exiting ones will be improved (e.g. to fully manage workflows).

3. Database related topics

3.1. Pages

3.1.1. Database requirements

InDesign documents are stored in Hermes as graphical objects, identified by a new object type (e.g. INDD_obj): the database entry (as for any graphical object) will have a link to the EPS file generated by InDesign, the JPG preview and the native INDD file.

An InDesign document is always associated with (and paginated in) a logical page, created automatically when the document is saved in Hermes. To enforce the similarity with current NewsRoom behaviour, InDesign will always access pages in the Hermes database, and will not access directly the associated object (created only for Hermes specific purposes).

The plugins for InDesign and InCopy have to handle these object types accordingly. InDesign needs to query the database for objects of type

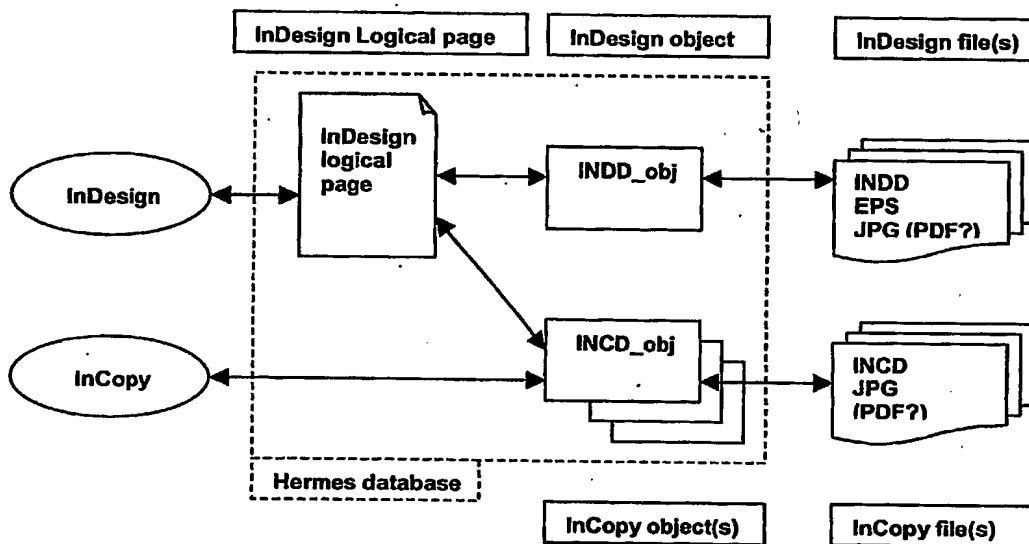
INDD_obj for opening pages, and for objects of type INCD_obj for paginating InCopy documents.

3.2. Texts

3.2.1. Database requirements

InCopy documents are stored in Hermes as graphical objects, identified by a new object type (e.g. INCD_obj): the database entry (as for any graphical object) will have a link to the JPG preview and the native INCD file. When an InCopy document is paginated in an InDesign logical page, the link information has to be stored and accessible in the database. Objects of type INCD_obj should not be visible or accessible from NewsRoom.

The following diagram summarizes the database items and their relationships.



UPS Explorer should provide functionality to open both object types (INDD, INCD) in the corresponding application.

3.3. Importing NewsRoom texts

Using the "Hermes palette" described above, it will be possible to import texts created with NewsRoom in InCopy.

There are several options available in order to transfer textual content from NewsRoom to InCopy:

- **Transfer texts as raw ASCII files, removing all the formatting applied:** this option will not be acceptable in most of the cases;
- **Transfer texts converting NewsRoom typographical commands to InDesign markup language:** although technically feasible (InDesign markup language is apparently very similar to the one of NewsRoom) this would

- require a huge effort; probably the translation should be limited to the most important and frequently used commands (e.g. <CF>, <CP>, etc.);
- **Transfer texts converting NewsRoom tags to InDesign styles:** this is probably the easiest solution, but it is also quite effective. This conversion does not require any interpretation of the tag definition, but simply includes tag names in the exported text, following the syntax required by InDesign to represent styles: when in InDesign or InCopy, formatting defined within styles will be applied to tagged text. As an enhancement of that, it is possible to consider a mapping mechanism, to allow a customisable mapping of NewsRoom tags to InDesign styles: the definition of the mapping will be saved to be re-applied during the import process.
- **Transfer texts converting NewsRoom tags to XML tags:** this is the most generic solution, and should be taken into account with the new XML support available in InDesign/InCopy Rel. 2.0. Again, this conversion does not require any interpretation of the tag definition: texts are exported with NewsRoom tags converted to XML tags (some elaboration of the tag order may be required to obtain well-formed XML documents). InDesign and InCopy have the capability of importing XML files, and to automatically associate styles to tags (mapping has to be pre-defined): the end result will be the same described in the previous point, but the process will be more generic. It is important to point out that the XML representation of Hermes objects available in Rel. 6.0 is far too complex for InDesign, that is not capable of managing such a rich document structure: therefore, the XML extracted from Hermes will have to be simplified in order to be imported successfully in Adobe tools.

3.3.1. Preserving text features

3.3.1.1. Revision tracking

InCopy 2.0 implements the concept of revision tracking with the "Changes" feature. The feature is very close to NewsRoom revision tracking: in fact, when importing texts created with NewsRoom, it should be possible to maintain revision info, by converting them to the Adobe format.

3.3.1.2. Notice Mode

InCopy and InDesign 2.0 implement the concept of Notice Mode with the "Notes" feature. The feature is very close to NewsRoom notice mode: in fact, when importing texts created with NewsRoom, it should be possible to maintain all the portions in notice mode, by converting them to the Adobe format.

3.3.1.3. NewsRoom Versions

Not surprisingly, there is no equivalent in InCopy or InDesign of the NewsRoom capability of managing different versions of the same story: the history of all the versions will be maintained, in Hermes, storing multiple copies of the same object in the database.

3.3.1.4. Tables

InDesign and InCopy 2.0 include powerful options for creating complex tables. Basic features are exactly the same as in NewsRoom (e.g. Merge/Split cells, rotate text, cell border and fill), but Adobe tools are more sophisticated due to OpenType support and advanced import capabilities. Anyway, it should be possible to import table

content created using NewsRoom from the Hermes database into InDesign/InCopy, with the same approach presented above for texts.

3.4. Importing graphical content

Using the Hermes palette, it should be possible to import graphics from Hermes in InDesign: the operation will be a link to a graphic container, equivalent to InDesign "File/Place" menu command. InCopy does not have any capability for managing graphic objects.

The import process will follow the rules and – eventually – the restrictions set by InDesign: in Hermes, graphic objects can be saved in the database using Photoshop plugin, NewsCrop application of HermImport server tool. In any case, the high-resolution file is stored in the associated filesystem, together with a low-res and a medium-res version of the graphic: all the image formats supported by Hermes (JPEG, TIFF) are supported by InDesign as well.

InDesign also supports EPS and DCS files, and OPI links eventually included: there is also a useful pre-flight function that could be invoked when generating the EPS version of a page.

In particular, when importing EPS files, several import options are available in InDesign:

- Read Embedded OPI Image Links
- Apply Photoshop Clipping Path
- Proxy Generation

3.5. Photoshop images

Currently Photoshop is integrated with Hermes, with a set of plugins enabling the application to save and retrieve images from the database: this set of functionalities has to be maintained and supported in the new integration scenario.

3.6. Illustrator objects, PDF files

InDesign natively supports both Illustrator and PDF documents, with several options available when importing them. This native support will always be available as a possible way for supporting these formats, but it is necessary to evaluate the possibility of storing these objects in the Hermes database, and to paginate them in InDesign from there. This implies creating new Hermes object types (e.g. Illustrator object, PDF object), and therefore will increase the complexity of the integration: however, there will be evident tremendous benefits for being capable of supporting the Adobe suite in Hermes.

It is not strictly necessary to integrate all the applications with dedicated plugins: for tools used less frequently, a server-based import process (e.g. evolution of HermImport), watching several folders waiting for files to be processed may be an acceptable solution. This approach may eventually adopt Adobe AlterCast to convert or generate different versions of incoming files (e.g. to generate an EPS file from a .ai file).

4. Hermes configuration

Object types: two new object types need to be introduced.

Levels: no changes, but it will be required that levels devoted to contain InDesign Logical pages will be allowed to contain objects as well. This to store both the page and the required object in the same place, in order to simplify the entire process.

Permissions: the following items need to be updated in order to manage two new object types: **Access tree classes**, to set permissions on objects in specific levels (create, read, write, etc.); **Access Object Classes**, to set the initial status for new objects.

Statuses: it appears there is not a specific need for changing statuses management. Both object and logical page statuses have enough options to cover the needs of InDesign / InCopy objects. As an example, the attribute "All objects ready" available for logical pages could be used to trigger the generation of EPS and the lock of all the content in a page.

Functional permissions: it is possible to envisage a set of permissions for InDesign and InCopy, to further customize user profiles by inhibiting or not program functionalities. They will be associated to workstations/permissions groups as it is today with Hermes client applications.

5. What's left

Although this integration has been thought through from a marketing perspective, there are still technical evaluations to be done. In detail, they are

- **Hermes Web Agent:** It is required to verify, if the functionality required by this integration is available or need to be developed
- **OPI:** Technical evaluation, whether the UNISYS OPI process can handle all possible PS output from InDesign
- **Import and Export procedures**
- **AdInterface**
- **Maintenance**



Advanced Query in Third Party Applications Design Specification

HE70-DS-AQTPA.doc

Authorized for issue by:

G Colagrossi

**Unisys
Integration Plan
Project Manager**

Signature

Date

A Politi

**Unisys
Adobe Integration and
Integration Platform
Project Manager**

Signature

Date

**January 21, 2003
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Advanced Query in Third Party Applications - Design Specification				
Author	A. Politi Integration Platform and Adobe Integration Project Manager Unisys				
To Be Approved By	AP Integration Platform and Adobe Integration Project Manager Unisys				
Comments	This is the Design Specification for the Advanced Query in Third Party Applications project				
File Name	HE70-DS-AQTPA.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1	All	New	01/22/2003	BC	First version



Table Of Contents

1	SCOPE	5
1.1	SYSTEM OVERVIEW	5
1.2	DOCUMENT OVERVIEW.....	5
2	REFERENCED DOCUMENTS	5
3	SYSTEM DESIGN DECISIONS.....	6
3.1	HIGH LEVEL SYSTEM BEHAVIOR.....	6
3.1.1	<i>Description of advanced query</i>	6
3.2	LIMITATIONS.....	8
4	PERFORMANCE.....	9
4.1	SPEED	9
4.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY.....	9
4.3	CAPACITY	9

1 Scope

1.1 System Overview

The purpose of this project is the integration of third parties applications with the Hermes editorial system. Third party applications are required to support an advanced query mechanisms that behaves like the Unisys Hermes Explorer application.

1.2 Document Overview

This document describes how the Integration Platform will implement the required services to allow third party applications to perform queries in an extended way, thus allowing users of Hermes client and external applications to have the same capabilities on both sides.

2 Referenced documents

HE70-FS-TPIN.DOC	Third Parties Integration with NewsRoom
HE70-DS-AIES.doc	Adobe Integration with eEditorial Solutions Project
HE70-FS-POMEA	Pages and Objects Management in External Applications
HE70-FS-AQTPA	Advanced Query in Third Party Applications

3 System design decisions

External applications that are integrated within Hermes produce different types of content according to their capabilities.

The different contents produced are standardized to pages and object inside Hermes. By taking into consideration the standardization, Hermes will automatically handle queries on both internal and external pages.

In designing this feature, we will consider the current system architecture to obtain maximum performances and to create a powerful query SOAP engine that makes maximum use of the capabilities offered by Hermes. Since the implementation of the Hermes query is powerful enough to support this implementation, the query via SOAP will be a wrapper around the Hermes API.

3.1 High level system behavior

The query in Hermes can be divided into object query and page query. All the search parameters available in Hermes will be available also to the SOAP integration platform.

3.1.1 Description of advanced query

Advanced query allow third party applications or, in general SOAP aware client applications, to search for objects and pages by specifying different criteria. All the search parameters are handled by placing an operator AND among all the values.

The query created on the client application is then built as a SOAP request that is sent for processing to the SOAP application server, which executes the query, and returns a resultset as a SOAP response.

Since Hermes is capable of storing thousand of objects and pages, a particular interest is placed on the performances of the query via SOAP.

Protocol restrictions to performances

Query executed via SOAP might be slower than query executed in the native Hermes environment because of the different protocols used. In the Hermes environment, a proprietary transport protocol over TCP is used and the protocol has been designed to be fast by reducing at minimum the verbosity of the transmission.

The HTTP protocol, instead, is designed to be a general purpose transport layer to move myriad of data format over the internet. To allow this, the HTTP is a verbose protocol with a lot of decorations of each packet.

When files are attached to the message, the SOAP technology moves packet in form of textual requests and responses, only in particular cases. Furthermore, the SOAP technology relies on XML, which uses a lot of "space" to describe information. Due to these considerations, each XML exchange should be reduced at minimum. For example, avoiding carriage returns in the response can dramatically decrease the size of the packet by keeping the validity of the XML payload. A simple carriage return is not significant itself but in the query response context which could be very huge, it makes the difference.

Because of the HTTP design, the implementation of the query must prevent the connection from being too long. For achieving this goal and considering the potential amount of query results returned by the Hermes server, a "maximum hint" rule must be implemented.

If a query result on the server is greater than a maximum allowed, the query must be refined in order to decrease the available record count. The optimum number is determined by taking a complex query result and considering a low speed connection that is used to interact with the SOAP server. Since the SOAP server will rely on STL for the implementation, it is recommended to pre-allocate a good chunk for the string size in order to avoid reallocations by the STL implementation.

STL uses an algorithm to resize the string which doubles the size each time a string need to be resized. By reserving a good amount of bytes on the string, the number of reallocation decreases dramatically and the speed for the XML string construction to be sent to the client will increase exponentially.

External objects

To support the external objects searches, the SOAP request must contain a tag that specifies which type of object to search for. However, the type of object different from Hermes types, cannot be determined "a priori" since the Integration Platform allows every application to be integrated without intervening in the Hermes software, a SOAP method to query the "currently supported object types" must be available to client applications.

Furthermore, to avoid fancy object types, each object type will be registered in an XML registry of object types by using a descriptive string and the MIME type of the application that generated that particular type of object.

The SOAP Integration Platform query service will then use the MIME type to perform the searches, while the client application will use the descriptive value to display a user-friendly choice to the user.

3.2 Limitations

No particular limitations are in place. All behaviors of the standard Hermes client query will be available through the SOAP integration platform.

4 Performance

Particular focus must be kept on the optimization. A prototype implementation has proven that with simple contrivances the performances of the query (both receiving and responding) will lead to an overall performance that is almost equal to the Hermes implementation.

4.1 Speed

No degradation in speed due to the implementation of the system.

4.2 Reliability, availability and maintainability

No degradation in reliability, availability and maintainability due to the implementation of the system.

4.3 Capacity

No degradation in capacity due to the implementation of the system.



SOAP Integration Platform Project Design Specification

HE70-DS-SIP.doc

**Authorized for issue
by:**

G Colagrossi

**Unisys
Integration Plan
Project Manager**

Signature

Date

A Politi

**Unisys
Integration Platform
and Adobe
Integration
Project Manager**

Signature

Date

**October 15, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

**Copyright © 2002 Unisys Corporation
All rights reserved**

Project Information

Document Title	SOAP Integration Platform Project Design Specification				
Author	A. Politi Adobe Integration and Integration Platform Project Manager Unisys				
	B. Merlo Analysis and Design Team		G.Galli Analysis and Design Team		
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys				
Comments	This is the Software Design Specification for SOAP Integration Platform				
File Name	HE70-DS-SIP.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	10/15/2002	AP	First version of the Design Specification
1.0	All	Update	19/12/2002	BM	Added SAS spec
1.0	All	Update	10/01/2003	AP	Global revision. Added collaboration diagrams

Table of Contents

1	INTRODUCTION.....	5
2	REFERENCES.....	5
3	DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	6
4	SYSTEM OVERVIEW.....	7
5	ARCHITECTURALLY SIGNIFICANT DESIGN PACKAGES.....	7
6	DESIGN CONSIDERATION.....	8
6.1	XML PARSER.....	8
7	ARCHITECTURAL GOALS AND DEPENDENCIES.....	9
8	CONSTRAINT.....	10
8.1	HARDWARE CONSTRAINT AND REQUIREMENTS.....	10
8.2	SOFTWARE CONSTRAINT AND REQUIREMENTS.....	10
8.3	PERFORMANCES.....	10
8.4	NETWORK COMMUNICATION.....	11
8.5	VERIFICATION AND VALIDATION.....	11
9	SYSTEM LOGICAL VIEW.....	12
10	SUBSYSTEM DESIGN.....	12
10.1	SOAP APPLICATION SERVER.....	12
10.2	SOAP SERVICE LAYER.....	12
10.2.1	Service Registry.....	12
10.2.2	Service entry point.....	14
10.2.3	Parameters Map.....	16
10.2.4	Attachments parameters.....	17
10.3	HERMES WEB SERVICES.....	18
10.3.1	Module caching.....	18
10.4	WSDL.....	19
10.5	SOAP RESPONSE.....	19
11	METADATA MANAGEMENT FOR ADOBE APPLICATIONS.....	20
12	EVOLUTION.....	20

1 Introduction

This document is the detailed design specification of the SOAP Integration Platform system. It refers to the design phase of the project 70-IPAI-SoftwareDevelopmentPlan.

2 References

The following table shows the references to the standards used during the design.

Table 1 Referenced standards

Description	Reference
SOAP Specification version 1.3	http://w3.org/soap
OpenSSL specification and downloads	http://www.openssl.org
Multipart messages	http://w3.org
IANA official registered mime types	http://www.iana.org
OPI 1.3 specification	http://partners.adobe.com/asn/developer/pdfs/tn/OPI_13.pdf
OPI 2.0 Specification	http://partners.adobe.com/asn/developer/pdfs/tn/5660.OPI_2.0.pdf
Adobe XMP technology specification	http://partners.adobe.com/
The Unified Modeling Language specification version 2.0	http://www.rational.com

Reader should also refer to the following Unisys documents

Table 2 Unisys referenced documents

Description	Reference
HE70-DS-SAS	<u>SOAP Application Server Design Specification</u>
HE70-FS-AQTPA	Advanced Query in Third Party Applications via SOAP Integration Platform
iPlatModel..mdl (temporary name)	SOAP Integration Platform Rose Model
HE70-MS-AINCM.doc	Vision document – Solution Management requirement for third party application
HE70-DS-AIES.doc	Adobe Integration with eEditorial Solutions Design Specification

3 Definitions, acronyms and abbreviations

Table 3 Definition, acronyms and their references

ACRONYM	DEFINITION	REFERENCE
SOAP	Simple Object Access Protocol	http://w3.org/SOAP
XML	Extensible Markup Language	http://w3.org/XML
HTTP	Hyper Text Transmission Protocol	http://w3.org/Protocols
HTTP-S	Hyper Text Transmission Protocol – with data encryption	http://w3.org/Protocols
IFRA	largest international exhibition dedicated to newspaper and media technology	http://www.ifra.com
Hermes Agent	An HTTP bridge to the Hermes system	
CMMI	Capability Maturity Model Integrated	http://www.sei.cmu.edu/cmmi/
DOM	Document Object Model	Standard for XML navigation, parsing and construction http://w3.org/XML

4 System Overview

The SOAP Integration platform is a framework built on top of the existing Hermes editorial system. The purpose of this software layer is to make a large set of Hermes API available to system integrators, in order to produce content with different software that can take advantage of the Hermes editorial production system.

The design of the system relies on standard protocol for the communication and on SOAP as the lightweight protocol to enable interaction with the outside world.

Particular interest must be reserved to extensibility and modularity, in order to minimize the impacts on other components when adding functionalities. The system will be separated into two logical components:

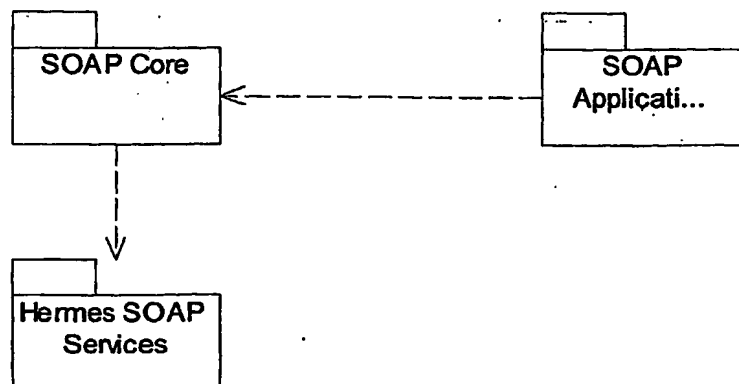
- Core Service, responsible for managing the run-time mapping from SOAP calls to WEB services and vice versa
- WEB Services, responsible of wrapping Hermes calls in a transparent way to the Core Service layer. Each request dispatched from the Core Service is rolled out by each specific WEB service.

The entire architecture includes a new Hermes module responsible of managing HTTP connections/request.

5 Architecturally Significant Design Packages

An overall component diagram of the system is shown below:

Figure 1 SOAP Integration Platform overview



6 Design Consideration

Most of the code used to write Hermes is C/C++ code and, although the core has a well-defined set of APIs, it does not allow direct interaction with applications different from the clients that are part of the system.

Furthermore, Hermes uses a proprietary communication protocol that is difficult to “externalize”, because it requires that a set of specific libraries on server and on the client must be available.

This requires that the client must be equipped with a software package that prevents it from being completely independent from the implementation of the server side.

The design of this framework will focus primarily on extensibility and modularity:

- **Extensibility:** by separating the core service module from the actual implementation of wrappers around Hermes API, the extensibility is achieved, since is implicit the ability of adding new services.
- **Modularity:** in order to achieve the maximum level of extensibility of the framework, the core service layer must be separated from the actual Hermes wrapper implementation and specific services will be provided by small components plugged to the framework.

The design of a SOAP Platform that lies on top of Hermes is possible without a great effort because of the similarity of the network protocol used by Hermes with the HTTP concept. In fact, although they are different in constitution, both rely on the request response strategy to implement communication between client and server.

6.1 XML Parser

XML is managed through the use of the Xerces library, which offers a C++ implementation of the DOM conform to the W3C specification.

7 Architectural Goals and Dependencies

The SOAP Integration Platform will be designed to exploit the existing Hermes implementation, allowing the Hermes system to interface with any other external application. The WEB services will implement services by calling and arranging the existing Hermes APIs. Specific assumption and dependencies are:

- There will be no new development inside Hermes to adjust gaps between SOAP and the core. Instead, the design assumes as a fixed point the existence of Hermes.
- A new module will be implemented as the endpoint server for two reasons:
 - The existing hermesagent cannot be overloaded since it is used for the HermesWeb product and no modification to the performances is possible.
 - The Integration Platform is a product that Unisys will deliver under license and not as an add-on to an existing licensed product.
- No direct database operation will be performed. Each access to the Hermes storage will rely on the existing Hermes APIs, in order to leverage the current capabilities.

8 Constraint

As mentioned before, Hermes is a proprietary system built with C/C++ language. Java should be the natural way of building a SOAP extensible layer but the gap between the two languages is difficult to be managed and, when managed, it relies on a difficult to understand mechanism that cannot be easily reproduced.

8.1 Hardware constraint and requirements

The SOAP integration platform will be available for both Intel and SPARC processor.

8.2 Software constraint and requirements

For the software environment:

- Windows 2000 Advanced Server
- SUN Solaris 5.8, 32 bit version

Both platforms will be available according to the Unisys certified standards. The implementation of the code must take into consideration of the architectural differences among platforms.

8.3 Performances

The design of the framework will take into high consideration the performance issue and no degradation or limitation to the existing Hermes system will be caused by the framework itself.

8.4 Network Communication

Network communication between the Integration Platform and the various actors will rely on standard HTTP / HTTPS transport over TCP/IP protocol. The communication mechanism used inside Hermes is request-response based. Client applications post a request to the server and wait for the response from the server which does all the processing requirement. The same concept is applied to the HTTP, with the main difference in the "purpose" of the protocol. The Hermes proprietary is tailored for the specific application needs, while the HTTP is a multi-purpose generic protocol.

This means that performances are quite different, since the HTTP needs a lot of extra meta-information to describe the data it is carrying over the network. Due to this verbosity, a particular attention is paid in the optimization of long responses from the SOAP server.

The SOAP integration platform provides communications in both plain HTTP and over Secure Socket Layer by using the OpenSSL library. The secure framework implementation is, however, responsibility of the SOAP application server, which design is detailed in document HE70-DS-SAS.

8.5 Verification and validation

The entire platform will be tested with specific procedures to ensure that:

- Multiple concurrent SOAP messages can be delivered to the services
- Buffer overflow protection is correctly implemented
- Invalid XML packets are correctly handled without problems to the SOAP core
- Malformed HTTP requests are handled correctly without problems to the SOAP core

Each developed service will be placed under regression test to ensure that no alterations to the SOAP core service layer have been made.

The alpha tests will be made by using scripts to send XML calls to the framework. The scripts will be refined on use and delivered to the testing group as an add-on to help integrity test.

9 System Logical View

In order to depict a logical view for the system, a typical usage scenario is shown below.

10 Subsystem design.

10.1 SOAP Application Server.

The SOAP application server is described in document HE70-DS-SAS

10.2 SOAP Service layer

The SOAP Service Framework is responsible of the management of the SOAP requests received by the SAS. The SAS checks the HTTP header for validity of session id and SOAPAction values. If these values cannot be verified, the request is not handled and an error is returned to the caller in the form of a SOAP fault.

If the HTTP header is valid, the SAS calls the SOAPService Registry and passes the SOAP request. The SOAP service registry looks for the service handler (WEB Service) that implements the method requested via SOAP and, if a service is found, it dispatches the SOAP call to the specific WEB service for being processed. The SOAP core will offer a set of methods to the WEB services, thus allowing to:

- Get parameters from the SOAP call
- Get the value of the parameters appropriately converted to C/C++ type
- Create the SOAP response to the call and abstract any specific XML construction and handling

10.2.1 Service Registry

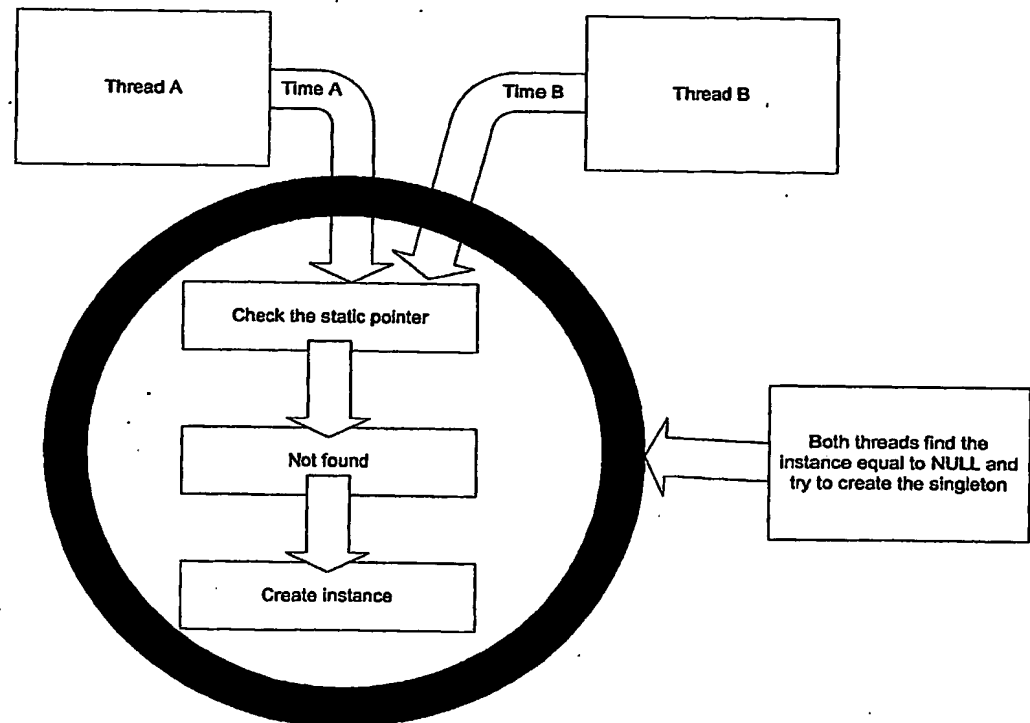
The service registry lies in the SOAP core and is responsible of maintaining the list of service and the mapping between SOAP methods and services that implement them. In order to keep a high level of extensibility, the services are mapped to specific calls via an XML file (deployment file) that is read during the server start up.

Doing so, enabling or disabling specific services will simply imply removing the mapping inside the deployment descriptor.

Furthermore, a particular protection level is placed into the service registry to allow continuing to offer services also if one or more services are deleted from the file system. The service registry is implemented as a singleton object using the Singleton pattern, in order to give the caller a single point of access to the WEB Service instantiation point.

The service registry singleton should be implemented with a double-checked locking pattern (Schmidt, 1 2001 145). Although the service core itself does not dispatch services into different threads because it relies on the multithread SAS engine, this rule could become invalid for performance reasons. By definition, the singleton exists in a single instance. The constructor of this type of object, however, cannot have a locking mechanism because of the following race condition:

Figure 2- Multithread unsafe singleton instantiation



Thread A

Thread B

Check the static pointer

No static pointer is available Check the pointer

Lock

No pointer available

When the thread A is in the LOCK condition is about to create the static pointer. When the thread B comes into, no instance is available and proceeds to its creation. To prevent this, the double-checked locking pattern [2] tests for the instance of the object and, if it does not exist, the object is locked. After the lock is acquired, the thread tests again the instance to see if in the meantime another thread created it. The maximum level of safety is achieved because the second thread waits until the mutex is available and, when this condition is true, the thread finds the instance already created and returns the pointer to the instance itself.

<< Concurrent access to the singleton – Drawing Placeholder>>

Figure 3 Thread safe singleton instantiation

After the Service registry has determined the method that is being called by parsing the SOAP request, it returns a pointer to the WEB service entry point in order to dispatch the work.

By using the double-checked locking pattern, the atomicity of the change to a singleton object is guaranteed even in multi CPU environment. In a single CPU environment, in fact, threads are “interleaved” and concurrency occurs less frequently than in multithreaded applications running on different CPUs.

The service registry is as an object factory and should be implemented as in [4] to decrease the interdependency between implementation of the core and implementation of the services and to speed up the SOAP dispatching.

10.2.2 Service entry point

In order to decrease the dependency between the core and the modules, the SOAP integration platform uses a deployment descriptor to map the service implementation to a service entry point. Whenever a request comes into, the service registry looks up the SOAP method into the XML payload and searches for a suitable implementation. If an implementation is found, its entry point is called to instantiate the actual service implementation.

C++ language does not allow a class to be created using its name as parameter, for example, to the constructor. A code such as

```
const char *className = "SpecificClass";  
  
BasePointer *instanceOfTheClass = new className;
```

Is not valid.

The solution of such a problem, which enables the ability of mapping service to method in an external file, is to use the inheritance to make all the service implementation specialization of a base class. Each implementation method inside the WEB service, has a C entry point whose name is actually mapped to the service via the deployment. The service registry reads this entry point from the deployment descriptor, loads the WEB service dynamically and gets the address of the creator function (entry point). Once this has been done, the service registry calls the entry point of the WEB service that does nothing but creating a new "class" and returning a pointer to it.

Implementation class

```
class SOAPImplementation : public SOAPServiceHandler
```

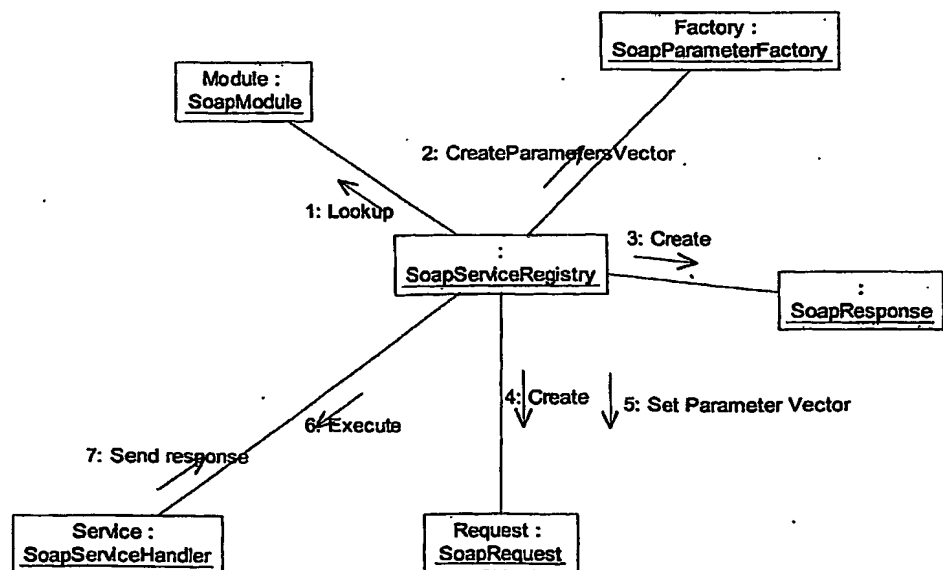
The entry point will be

```
SOAPServiceHandler *EntryPointFunction()
{
    return new SOAPImplementation;
}
```

This is the only way to bypass the previously mentioned C++ limitation. By using this technique, each SOAP call can be added to the entire framework without recompilation of the layer.

The collaboration diagram for the service lookup and dispatching is reported in Figure 4

Figure 4 Service instantiation and dispatching



The service registry maintains a mapping between the service name exported and the module that implements the function. This mapping is loaded at the service registry instantiation by reading the deployment descriptor from the file system. The deployment descriptor is an XML file that maps the name of the entry point inside the service and the dynamic library with the implementation. Since this mapping does not change during the service registry run time, it should be optimized for fast searches rather than for fast insertion. A suggestion for obtaining fast searches is to use the standard library vector instead of a map. As discussed in [3] using a sorted vector of elements is significantly faster than using a map of keyed value, since the map is implemented as a RB-tree with overhead that is optimized for fast insertion.

Once the service has been found and instantiated, the SOAP core finishes to parse the XML and loads the parameter map.

10.2.3 Parameters Map

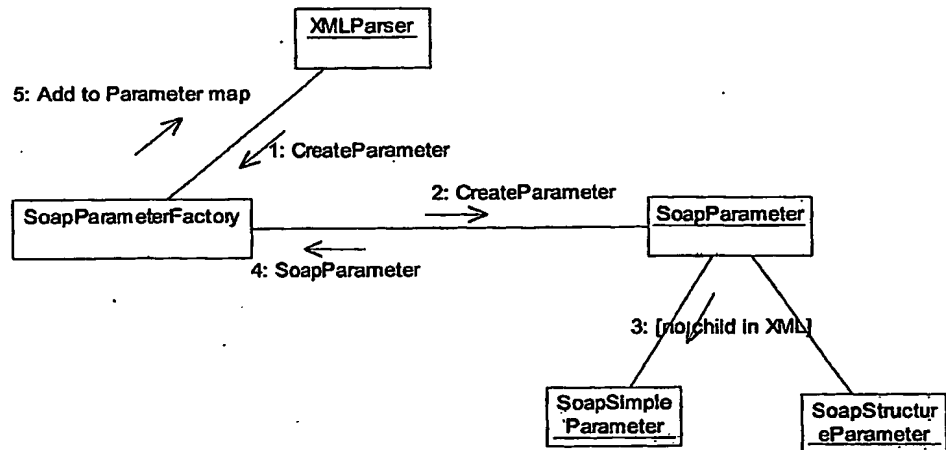
The SOAP core manages two types of parameters:

- Simple parameters
- Complex parameters (structure)

Simple parameters are used to map directly name-value pair, while complex parameters can contain simple parameters. This allows the SOAP framework to manage a large variety of parameters combination. In fact, by using a Composite Citations

[1] pattern, it is possible to manage parameter that can be composed of a structure and each element of the structure is composed by other structures. To create the correct parameter object, which depends on the XML structure of the SOAP request, a class factory Citations

[1] is used to delegate the construction of the appropriate object. The collaboration among object is represented by the following diagram:

Figure 5 Collaboration diagram for parameter creation

The parameter map should be implemented for fast retrieval rather than for fast insertion, thus the suggestion becomes the same as per the service mapping discussed in chapter 10.2.1

10.2.4 Attachments parameters

The integration platform allows creation of content with third party applications and such a content is "converted" to Hermes objects and pages. SOAP incoming messages must have the ability to attach files. To allow the mixed XML and binary content in a SOAP message, a multipart – mime message is used. The SOAP core is responsible for the parsing of the multipart message and the storage of the raw bytes of the attachments. Details of messages with attachments are provided in the HE70-DS-AIES document, where the interface between the system and client applications is fully described.

10.3 Hermes Web Services

Hermes Web Services are components that are plugged into the SOAP Integration Platform to perform the real request processing. These modules are implemented as dynamic link libraries that are loaded on demand and stay active during all the run time.

Hermes Web Services have the responsibility to wrap Hermes legacy code into object-oriented components that are instantiated by the platform. The effect on the Hermes code is null and the implementation can leverage the existing API, thus minimizing the effort to make Hermes an open system.

The services should be divided into "categories" depending on the task they run. A special category is that including services responsible of critical task, where this tasks cannot be unplugged from the platform since they are vital for the existence of the platform. In this category, the following services will be placed:

- Login service
- Logout service
- Session management service

These types of services cannot be mapped into the deployment descriptor, thus they cannot be unplugged from the platform.

Hermes specific services are divided into the following categories:

Service category	Hermes specific task
Objects	Hermes object management, such as creation, deletion, retrieval of objects.
Pages	Hermes page management, such as creation, deletion, retrieval of pages, link/unlink of objects to/from pages
Edition	Edition management, such as edition listing, zone management
Workflow	Object and page status management
Users	User management, such as user data retrieval, permission checking
Level	Level management, such as level browsing

10.3.1 Module caching

The SOAP core manages the various WEB services by keeping a table of method/service implementation. Each time a method is run, the table is updated by incrementing the usage count for the module and decrementing each module that has not been hit. When the counter for a module is equal to 0 (zero), the module is discarded thus gaining memory and, consequently, performances.



10.4 WSDL

Each implementation module should be able to describe itself and the parameters name/types, as well as the return values. This enables the SOAP core to implement a system service that collects the information about the services. By doing this, a discovery tool can be created to give a "generic client" a way to obtain information about what Hermes SOAP is offering in term of WEB services.

10.5 SOAP response

The responsibility for creating response is equally divided between the service handler (implementation of service) and service dispatcher. The service dispatcher offers a response object to the called implementation that, in turn, fills in the return value elements that compose the response. The service dispatcher creates an XML SOAP response by attaching the response built by the service implementation to the SOAP envelope. The service dispatcher is responsible to create the correct XML response and to verify that it is conformant to the SOAP specification.

11 Metadata management for Adobe Applications

Metadata will be handled whenever possible using the XMP technology. XMP is an Adobe proprietary technology platform for metadata embedding inside data file. The XMP usage has different advantages, such as:

- Free of charge and OpenSource: this allow the developer to work with the source code which is extremely important to understand the working internals and for the debugging of application
- Cross platform. The XMP SDK is delivered and tested on different platforms, such as UNIX and Windows. This enables the complete reuse of the core code
- Metadata are embedded into asset files using XML, Dublin Core specification.

Metadata can be embedded in InDesign/InCopy file directly from the application and extracted on the server, thus allowing the storage in a database that offers query functionalities.

XMP must not be used to embed metadata in image file. In fact, the JPEG specification for example, implies that a reader application or image manipulation software discards every non JPEG data inside the file. Opening a JPEG with XMP embedded metadata with software different from Adobe will result in removing the metadata.

12 Evolution

The architecture of the SOAP integration platform and the implementation based on legacy code wrapper, enables a high level of extensibility. After a first major release of the integration platform, new services will be added to perform even News Gathering Manager operations.

From the platform perspective, an interface for service management will be created, allowing the service startup and shutdown from remote client applications, such as an administration WEB page.

This design and architecture team will consider also a model for inter-service communication, to allow the reuse of code and implementation among services.

Citations

- [1] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley 1995
- [2] Douglas Schmidt, *C++ Network programming Volume 1 and 2*, Addison-Wesley 2001
- [3] Scott Meyers, *Effective STL*, Addison-Wesley 2001
- [4] Andrei Alexandrescu, *Modern C++ Design*, Addison-Wesley 2001



Pages and Objects Management in External Applications Project Functional Specification

HE70-FS-POMEA.doc

**Authorized for issue
by:**

U Gadarco

**Unisys
Project Manager**

Signature

Date

H Hofmann

**Unisys
Editorial Product
Manager**

Signature

Date

**November 26, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

**Copyright © 2002 Unisys Corporation
All rights reserved**

Project Information

Document Title	Pages and Objects Management in External Applications Project Functional Specification				
Author	BC Requirements Analyst Unisys				
To Be Approved By	UG Project Manager Unisys			HH Editorial Product Manager Unisys	
Comments	This is the Functional Specification for the Pages and Objects Management in External Applications project				
File Name	HE70-FS-POMEA.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	11/26/2002	BC	First version of the Functional Specification

Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	6
2.3	ASSUMPTIONS AND DEPENDENCIES	6
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE	6
3	FUNCTIONAL DESCRIPTION.....	7
3.1	MANAGING PAGES WITH THIRD PARTY APPLICATIONS	7
3.1.1	Description.....	7
3.2	MANAGING OBJECTS WITH THIRD PARTY APPLICATIONS	9
3.2.1	Description.....	9
4	INFORMATION FOR USER MANUAL	12
5	PERFORMANCE	12
5.1	SPEED	12
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY	12
5.3	CAPACITY	12
6	INSTALLATION	12



1 References

This functional specification refers to Pages and Objects Management in External Applications Editorial Enhancement EE-2002-03.

Refer also to the following documents:

- HE70-FS-TPIN functional specification for details on Third Parties Integration with NewsRoom
- HE70-FS-TPIHE functional specification for information about Third Parties Integration with Hermes Explorer.

2 Functional Overview

The functions described in this document aim at providing a general overview of how pages and objects created with third party applications will be managed in order to integrate them with the Hermes workflow. This is to make third party applications part of the suite of Hermes tools, allowing them to access content stored in the database, save pages and stories in the database, support workflows, user permissions and profiles.

2.1 Major functions

- Creation of logical pages using third parties applications, so that they can be integrated with the UPS editorial system.
- Save/Release of pages created with third parties applications in the Hermes database according to the UPS editorial workflow. Once stored in the database, it will be possible to take actions such as loading the page, deleting it, or use it for interacting with Hermes client applications such as UPS-Explorer and NewsRoom.
- Creation of objects using third parties applications, in order to save/release them to the Hermes database, so that they can be used both from UPS applications and third party applications (for example Adobe InDesign). Once stored, they can be searched through a specific query, loaded for further editing (into the native application only), deleted, and so on.
- Pagination (link) of objects (either created by third party application or by UPS applications) into external pages.
- Unlink of objects from pages.



2.2 Data/Activity Diagram

Not applicable.

2.3 Assumptions and Dependencies

The figures (if any) depicted in this document are dummy screenshots only.

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.



3 Functional Description

In order to support the integration of third party applications with the suite of Hermes tools, the client application will be provided with the capability to create logical pages in Hermes. Pages created with this tool will be different from those created with Newsroom and will be modified in their original format by the external application only. Other third party tools (for example Photoshop, Illustrator and InCopy, in case of Adobe applications) will be used for the creation of objects to be inserted in these pages. It will be possible to create a page first, then assign the various text containers to different users that will open the object they are assigned to and will write to fit. In this way, multiple users will be able to work on different objects contents present on the same page, while the page layout can be modified by a single user only. Pages and objects will be modified with the tools used for their creation.

Some of the functions described in this document will also be accessible through a tool palette that will be available in the external applications to permit the interaction between the client application and Hermes. The functions covered by this palette will be detailed in a dedicated document.

3.1 Managing pages with third party applications

3.1.1 Description

External documents (for example, pages coming from tools such as Adobe InDesign) will be stored in Hermes as logical pages. The corresponding database entry will have a link to the EPS file generated by the third party application, the JPG preview and the native external file.

Each external document will be a logical page that will be created when saving the document to Hermes. Once in the Hermes database, external logical pages will be available for interaction with Hermes client applications, such as UPS-Explorer or NewsRoom. The interaction between external pages and the Hermes applications above-mentioned will be dealt with in dedicated documents (refer to the HE70-FS-TPIHE and HE70-FS-TPIN respectively). Any change to the original external page will be performed in the native application only.

The following functions will be provided for page handling using the client application:

- **Create page.** This function will allow users to create a new external page based on the available page grids (according to the chosen level). The application will

display a dialog window requiring users to enter the page level. The system will then display a list of available editions according to the chosen level. If the selected level requires a publication date, the system will ask the user to enter it. In addition, users will have to provide a name for the page being created.

This action will result in the creation of a new record for a logical page in the UPS database and the locking of this record. An initial status will be set for the created page.

- **Save Page.** This function will allow saving the external page by displaying a dialog window where users can specify Level, Edition, Page name and publication date (if required by the selected level). The save operation will update the external record in the database. The process of saving pages requires the native external document to be sent by the client application.
Each time a save operation is performed, the client application will attach a preview in order to let Hermes display a preview in all other client applications.
- **Release Page.** To release the external page, users will select the appropriate status the page will be moved to according to the e-Editorial workflow from the dialog window that will be displayed when the Release function is selected. The Release dialog window will list the entries corresponding to the statuses along with a small square filled in with the color representing the status (as currently available in UPS client applications). The Release action will update the logical page record and the logical page will be unlocked. As for the save operation, the native external document will be sent by the client application. In addition, the client application is responsible for creating and attaching the relevant preview document if required by the workflow. If the page contains links to images files that come from the Hermes system, OPI comments will be placed in the EPS to reference the path of the high-resolution version of the image.
- **Load page.** When loading an external page from the database the page will be open in its native application. Users will then be able to modify the page according to their needs using the tools provided by the originating application. Usually, the load operation is performed after executing a query. Opening pages will also be possible through the Hermes tool palette which will be available into third party applications.
- **Delete page.** Page deletion will be allowed via a push-button, as well as via menu item. If the page is locked, the operation will fail. If the operation succeeds, the database will be immediately updated. Deleting pages will also be possible through the Hermes tool palette which will be available into client applications.

3.2 Managing objects with third party applications

3.2.1 Description

As for pages, it will be possible to create objects (texts or image type objects) with third parties applications and store them into the Hermes database. The corresponding database entry will have a link to the JPG/EPS preview and the native external file. When an InCopy object is paginated in an InDesign logical page, the information relevant to the link will be stored in the database. When these objects are loaded in order to be modified, they will be opened in their native application.

UPSExplorer will allow to open objects coming from both Newsroom and InCopy applications in the corresponding application.

In order to provide a tight integration between the Hermes system and third party applications, the following function will be available:

- **Create object.** This function will allow users to create a new object (either a text or an image object). The application will display a dialog window requiring users to enter the object level. The system will then display a list of available editions according to the chosen level. If the selected level requires a publication date, the system will ask the user to enter it. In addition, users will have to provide a name for the object being created.
This action will result in the creation of a new record for an object in the UPS database and the locking of this record. An initial status will be set for the created object.
- **Save Object.** This function will allow saving the third party object by displaying a dialog window where users can specify Level, Edition, Object name and publication date (if required by the selected level). The save operation will update the record in the database. The process of saving pages requires the native textual document to be sent by the client application. Each time a save operation is performed, the third party application will attach a JPEG preview in order to let Hermes display a preview in all other client applications.
- **Release Object.** To release the external object (or other third party object), users will select the appropriate status the object will move to according to the e-Editorial workflow from the dialog window that will be displayed when the Release function is selected. The Release dialog window will list the entries corresponding to the statuses along with a small square filled in with the color representing the status (as currently available in UPS client applications). The Release action will update the object record and the object will be unlocked. As for

the save operation, the native external document will be sent by the client application.

- **Load Object.** When loading an external object from the database (or an object created with another third party application) the object will be open in its native application. Users will then be able to modify the object according to their needs using the tools provided by the originating application. Usually, the load operation is performed after executing a query. Opening objects will also be possible through the Hermes tool palette which will be available into third party applications.
- **Delete Object.** Object deletion will be allowed via a push-button, as well as via menu item. If the object is locked, the operation will fail. If the operation is successful, the database will be immediately updated. Deleting objects will also be possible through the Hermes tool palette which will be available into third party applications.
- **Link object to page.** It will be possible to paginate objects in InDesign pages either via menu item or, after executing a query, from the query results. The selected object will be paginated, provided that it is not already linked to another page. If this is the case, a warning message will be displayed and the link operation will fail.
Files from the file-system will be placed into InDesign documents only after they have been inserted into the Hermes database. If a local file is placed, users will be asked to create the object in the Hermes database by providing the Level, Edition, publication date (if required by the selected level) and name of the object being created. If the creation of the object fails, no files will be placed into the page. If the creation succeeds, the high-resolution is uploaded to Hermes and a Hermes generated low-resolution is downloaded for placement using the object ID to link the asset to the page. If placed files are image files, appropriate OPI comments will be inserted into the EPS to reference the high-resolution path of the image on the Hermes file system. Furthermore the EPS will contain the OPI comments for the transformations applied to the image.
If a local text is inserted in page, a new InCopy object will be created and then placed in page.
Linking objects to pages will also be possible through the Hermes tool palette which will be available into third party applications.
- **Unlink object from page.** Objects can be unlinked from an InDesign page directly from the query result or via a menu option. When an object is unlinked from the page, it will be removed from the page and the database will be updated immediately. This will free the object and make it available for pagination.
Unlinking objects from pages will also be possible through the Hermes tool palette which will be available into third party applications.
- **Versioning.** Versioning applies only to objects. It will be possible to keep track of different versions of an object. Users will ask for an overview of an old version of an open object from the client application via menu item. As a result, a list of

version numbers will be displayed. The version list will be sorted in a descending order, or last-first. The user can select a version and open it in read-only mode. A new version of the object will be created whenever the object is released to Hermes.

- **Dummy text** – Third party application textual objects (for example InCopy objects) found in a status with an attribute different from READY FOR TYPESET will be shown as dummy text instead of the actual text, preserving the text styles and format. Depending on clients needs, a proper status attribute will be inserted.



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.



Hermes Palette Availability in External Applications

Project Functional Specification

HE70-FS-HPAEA.doc

**Authorized for issue
by:**

A Politi

**Unisys
Adobe Integration
and Integration
Platform Project
Manager**

Signature

Date

H Hofmann

**Unisys
Editorial Product
Manager**

Signature

Date

**February 03, 2003
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Hermes Palette Availability in External Applications Project Functional Specification				
Author	BC Requirements Analyst Unisys				
To Be Approved By	AP Adobe Integration and Integration Platform Project Manager Unisys			HH Editorial Product Manager Unisys	
Comments	This is the Functional Specification for the Hermes Palette Availability in External Applications project				
File Name	HE70-FS-HPAEA.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	02/03/2003	BC	First version of the Functional Specification



Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	6
2.3	ASSUMPTIONS AND DEPENDENCIES	6
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE.....	6
3	FUNCTIONAL DESCRIPTION.....	7
3.1	HERMES PALETTE FUNCTIONS.....	7
3.1.1	Description.....	7
4	INFORMATION FOR USER MANUAL	9
5	PERFORMANCE	9
5.1	SPEED.....	9
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY	9
5.3	CAPACITY	9
6	INSTALLATION	9

1 References

This functional specification refers to Hermes Palette Availability in External Applications Editorial Enhancement EE-2002-03.

Further references:

- HE70-FS-AQTPA for details on the search function in third party applications.
- HE70-FS-POMEA for details on the page and objects management in external applications.

2 Functional Overview

Availability of a tool palette in third party applications providing access to content stored in the Hermes database, according to the workflow and users permissions. The Hermes palette will be a fast and easy way to perform operations on objects and pages retrieved from the database directly from the query results list.

Most of the functions provided by the Hermes palette will also be available via menu item and will be described in dedicated documents.

2.1 Major functions

The tool palette will provide users with quick access to the following functions:

- Capability to query the Hermes DB from a third party application in order to search for stored pages previously created with the external application. The query can be narrowed down by specifying appropriate parameters that will allow to easily retrieve the page(s) being searched.
- Capability to query the Hermes DB from a third party application in order to search for stored objects previously created either with the external application, or using the Hermes client applications (NewsRoom). The query can be narrowed down by specifying appropriate parameters that will allow to easily retrieve the object(s) being searched.
- Capability to open objects/pages by dragging&dropping the retrieved items in the application main screen. This capability will also be provided through a menu item, or double clicking the item to be paginated.
- Capability to paginate retrieved objects in pages created with third party applications.
- Capability to unlink objects linked to a page
- Capability to delete retrieved objects and pages.
- Import content of NewsRoom textual object into an external textual object.
- Capability to move objects/pages among levels



2.2 Data/Activity Diagram

Not applicable.

2.3 Assumptions and Dependencies

- The figures depicted in this document (if any) are dummy screenshots.
- The GUI of all new dialog windows/controls implemented must be XP-like style.

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.



3 Functional Description

In order to support the integration of third party applications with the suite of Hermes tools, third party applications will be provided with the capability to access contents stored in the Hermes database through a palette. The Hermes palette will gather a set of functions which will allow users to specify a query for objects/pages and to take further actions after objects/pages retrieval.

3.1 Hermes palette functions

3.1.1 Description

The tool palette will provide access to the Hermes database and to the functions detailed below.

- **Search** - This function will allow users to perform a query on the Hermes database in order to search for objects and pages. Users will be able to specify the search criteria in order to narrow down the query, thus allowing a fast and easy retrieval of the items being searched. The following fields can be used as search parameters:
 - *Object/page name*
 - *Level*. In Hermes, up to 5 levels are available. This will be supported also when performing a query from external applications.
 - *Publication date*
 - *Author*
 - *Status*
 - *Type*. Users will be able to retrieve either external objects (pages created with the application that starts the query), or Hermes objects (texts, headlines, captions, summaries, charts, logos, graphs). As far as pages are concerned, users will be able to search for external pages only.
 - *Object/page comment*
 - *Edition*
 - *Creator/modifier name and creation/modification intervals*
 - *Workflow fields*. Users will be able to specify if the object/page being retrieved is used/not used, and the status in which the item is found.

As the search function is also available via menu item, details on the above-mentioned search criteria will be provided in the HE70-FS-AQTPA document.

After retrieval, query results will be displayed in the tool palette. Users will be able to choose which fields pertaining to the retrieved objects/pages will be displayed. These fields will be configurable.

In addition, users will be able to take further actions (which are described below) on the retrieved objects/pages directly from the query results list.



- **Open object/page** – This function will allow users to open the retrieved external item by dragging&dropping it in its native application main screen. The UPS database will be updated, so that the item is marked as “locked”. Users will then be able to modify the object/page according to their needs using the tools provided by the originating application.
- **Link object to page** - This function will allow users to paginate retrieved objects into external pages (pages created with the external application that started the query through the Hermes palette). Retrieved objects can be paginated into the external page directly from the query results. The selected object will be paginated, provided that it is not already linked to another page. If this is the case, a warning message will be displayed and the link operation will fail. It will be possible to paginate objects created with the external application (for example InCopy texts will be paginated into InDesign pages), or Newsroom objects. If a Newsroom text is linked to an external page, an appropriate conversion will take place (see *NewsRoom text import* function described below).
- **Unlink object** – This function will allow users to unlink an object previously linked to a page. This operation can be performed by selecting the object in the query results list and choosing the appropriate function. After the unlink action is taken, the object will be marked as “available” in the database.
- **Delete object/page** — This function will allow users to delete a retrieved page/object directly from the query results. After the object/page deletion, the UPS database will be updated, so that the object/page is marked as “deleted”.
- **Unlock object/page** – This function will allow users to unlock an object/page already in use by another user. This action will be taken directly from the query results list.
- **NewsRoom text import into an external object/page** – Users will be able to select a NewsRoom textual object retrieved with the Hermes palette and import the object content into an external object/page open in its native application.
For example, when dragging the retrieved NewsRoom textual object into InCopy, the external application will generate a new InCopy object and import a flat version of the Newsroom text, without markup commands and text styles and convert the text into a new external object.
When users drag and drop a Newsroom textual object into an external page (for example an InDesign page), a new external text object (InCopy, for example) will be created with the content read from the Newsroom object.
- **Copy/Move objects/pages among levels** – This function will allow users to select an object/page from the query results list and copy or move it to a level different from that where it is currently found. Users will be able to select a new destination level from a dialog window that will be displayed when selecting this function.



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.



Third Parties Integration with Hermes Explorer

Project Functional Specification

HE70-FS-TPIHE.doc

**Authorized for issue
by:**

U Gadarco

Signature

Date

**Unisys
Project Manager**

H Hofmann

Signature

Date

**Unisys
Editorial Product
Manager**

**December 13, 2002
Version 1.0**

Unisys Proprietary Data Notice

The information contained in this document is proprietary to Unisys Corporation. The information in this document shall not be reproduced, shown or disclosed outside the Unisys Corporation without written permission. Information contained in this document is highly sensitive and of a competitive nature.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license or agreement to purchase services or equipment. The only warranties made by Unisys, if any, with respect to the products, programs or services described in this document are set forth in such license or agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document, including but not limited to direct, indirect, special or consequential damages.

You should be very careful to ensure the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with the respect to which it is used. The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

TEAMmethod is a registered service mark of Unisys Corporation.
Unisys is a registered trademark of Unisys Corporation

Project Information

Document Title	Third Parties Integration with Hermes Explorer Project Functional Specification				
Author	BC Requirements Analyst Unisys				
To Be Approved By	UG Project Manager Unisys	HH Editorial Product Manager Unisys			
Comments	This is the Functional Specification for the Third Parties Integration with Hermes Explorer project				
File Name	HE70-FS-TPIHE.doc				
REVISION HISTORY					
Rev	Section	Type	Date	Modified by	Remarks
1.0	All	New	12/13/2002	BC	First version of the Functional Specification

Table of Contents

1	REFERENCES	5
2	FUNCTIONAL OVERVIEW.....	5
2.1	MAJOR FUNCTIONS	5
2.2	DATA/ACTIVITY DIAGRAM	5
2.3	ASSUMPTIONS AND DEPENDENCIES	6
2.4	MIGRATION, COMPATIBILITY AND CO-EXISTENCE.....	6
3	FUNCTIONAL DESCRIPTION.....	7
3.1	SEARCHING FOR EXTERNAL OBJECTS	7
3.1.1	Description.....	7
3.2	SEARCHING FOR EXTERNAL PAGES	9
3.2.1	Description.....	9
3.3	OPENING RETRIEVED EXTERNAL ITEMS.....	10
3.3.1	Description.....	10
4	INFORMATION FOR USER MANUAL	11
5	PERFORMANCE	11
5.1	SPEED.....	11
5.2	RELIABILITY, AVAILABILITY AND MAINTAINABILITY.....	11
5.3	CAPACITY	11
6	INSTALLATION	11



1 References

This functional specification refers to Third Parties Integration with Hermes Explorer project.

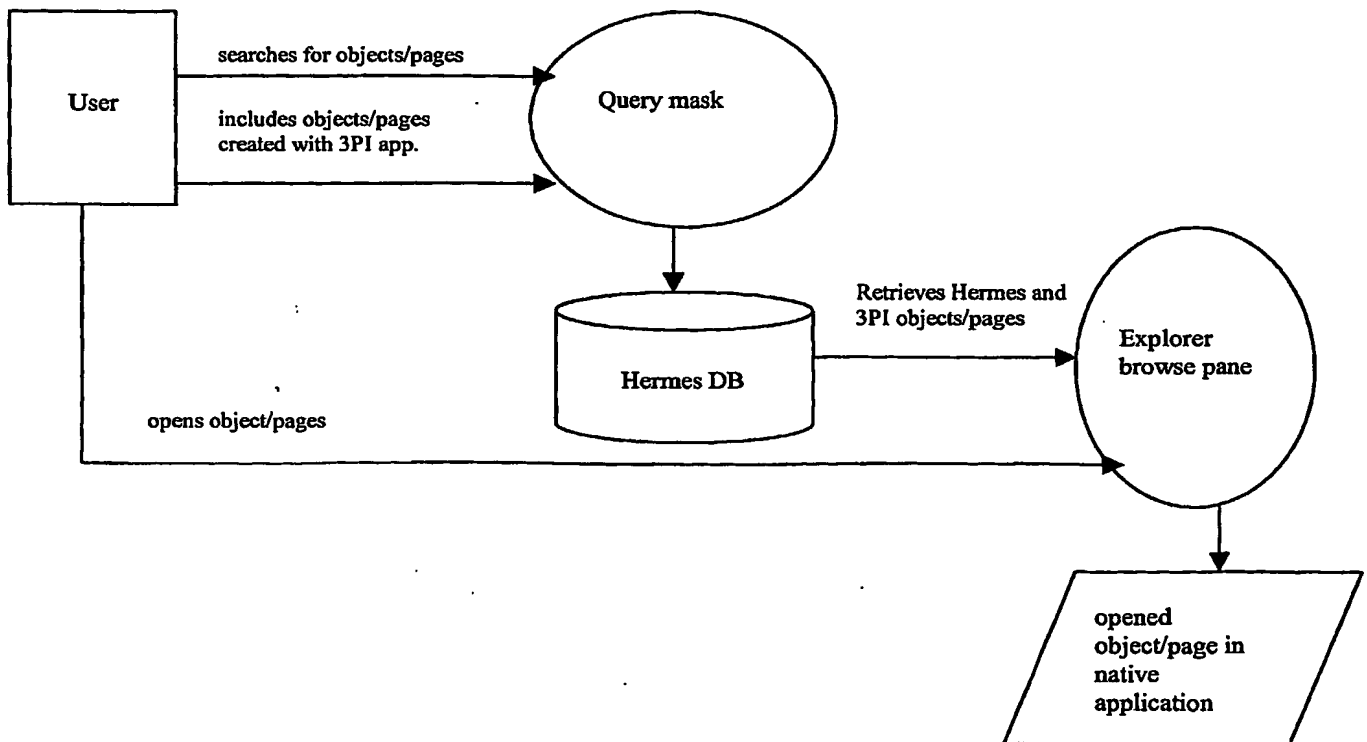
2 Functional Overview

The integration of third parties applications with the Hermes applications will imply some changes in the Hermes Explorer module, in order to allow queries on the Hermes database to retrieve objects/pages created with external applications.

2.1 Major functions

- Capability to query the Hermes database to search for objects/pages created with third parties applications.
- Capability to open the retrieved external objects/pages through the standard opening methods provided by the Hermes Explorer module.

2.2 Data/Activity Diagram





2.3 Assumptions and Dependencies

- The figures depicted in this document are dummy screenshots.
- The GUI of all new dialog windows/controls implemented must be XP-like style.

2.4 Migration, Compatibility and Co-existence

This functionality will be implemented as part of the overall Unisys Publishing Solutions environment.

New functionality must remain compatible with prior versions of software.

3 Functional Description

The integration of third parties applications with the Hermes editorial systems requires the Hermes modules to be partially modified in order to interact with external applications.

This project aims at providing users with the capabilities to query the Hermes database using the Explorer module in order to retrieve object/pages created with external applications (for example: Adobe InDesign and InCopy). The retrieved objects/pages can then be opened using the Hermes Explorer standard opening methods.

3.1 Searching for external objects

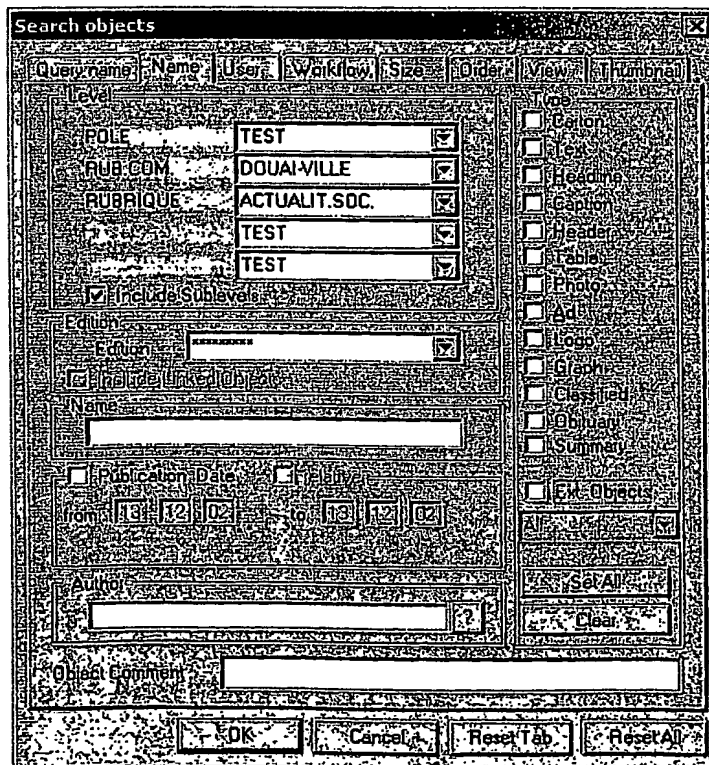
3.1.1 Description

Users will be able to query the Hermes database using Hermes Explorer to search for objects created with third parties applications.

This will imply the capability to specify the object type to be retrieved in the search criteria. To achieve this, the **Search Objects** dialog window will be modified.

In addition to the standard object types listed in the **Name** tab, a check box will be available, so that users are able to specify whether they want to retrieve external objects or not.

Figure 1 – External object types availability for objects query



By checking this box, a combo box will be enabled listing the external object types according to their native application. Users will have the capability to select the object type coming either from specific external applications (multi-selection allowed) or from all the third parties applications integrated with the Hermes editorial system.

Figure 2 – Selecting external object types to be retrieved

Of course, it will be possible to retrieve both Hermes objects and external objects.

3.2 Searching for external pages

3.2.1 Description

Users will be provided with the capability to search for pages created with third parties applications using Hermes Explorer.

To achieve this, the *Name* tab of the *Search Pages* dialog window will be modified, so that it is possible to specify whether internal pages (created with Hermes) or external pages (created with third parties applications) have to be retrieved from the Hermes database. Of course, it will be possible to search for both internal and external pages, or just one of these types.

Figure 3 – Availability of internal and external pages type to be specified in the search criteria

The screenshot shows the 'Search pages' dialog box with the 'Name' tab selected. The 'Internal Pages' and 'External Pages' checkboxes are both unchecked. The search criteria section includes a 'Level' tree view with 'FOLE', 'RUB.COM', and 'RUBRIQUE' visible. Below this are four search fields: 'TEST', 'DOUAI-VILLE', 'ACTUALIT.SOC.', and 'TEST', each with a dropdown arrow. There are also checkboxes for 'Include Sublevels' and 'Include virtual pages'. A 'Name' field is present. At the bottom, there are checkboxes for 'Page and Reference' and 'Publication date', along with 'from' and 'to' date pickers. A 'Page comment' field is at the bottom. Buttons for 'OK', 'Cancel', 'Reset Tab', and 'Reset All' are at the bottom right.

If "Page and Reference" is specified as search criteria for external pages, the query results list will show the retrieved external page, along with all the objects (both external and Newsroom objects) linked to this page.

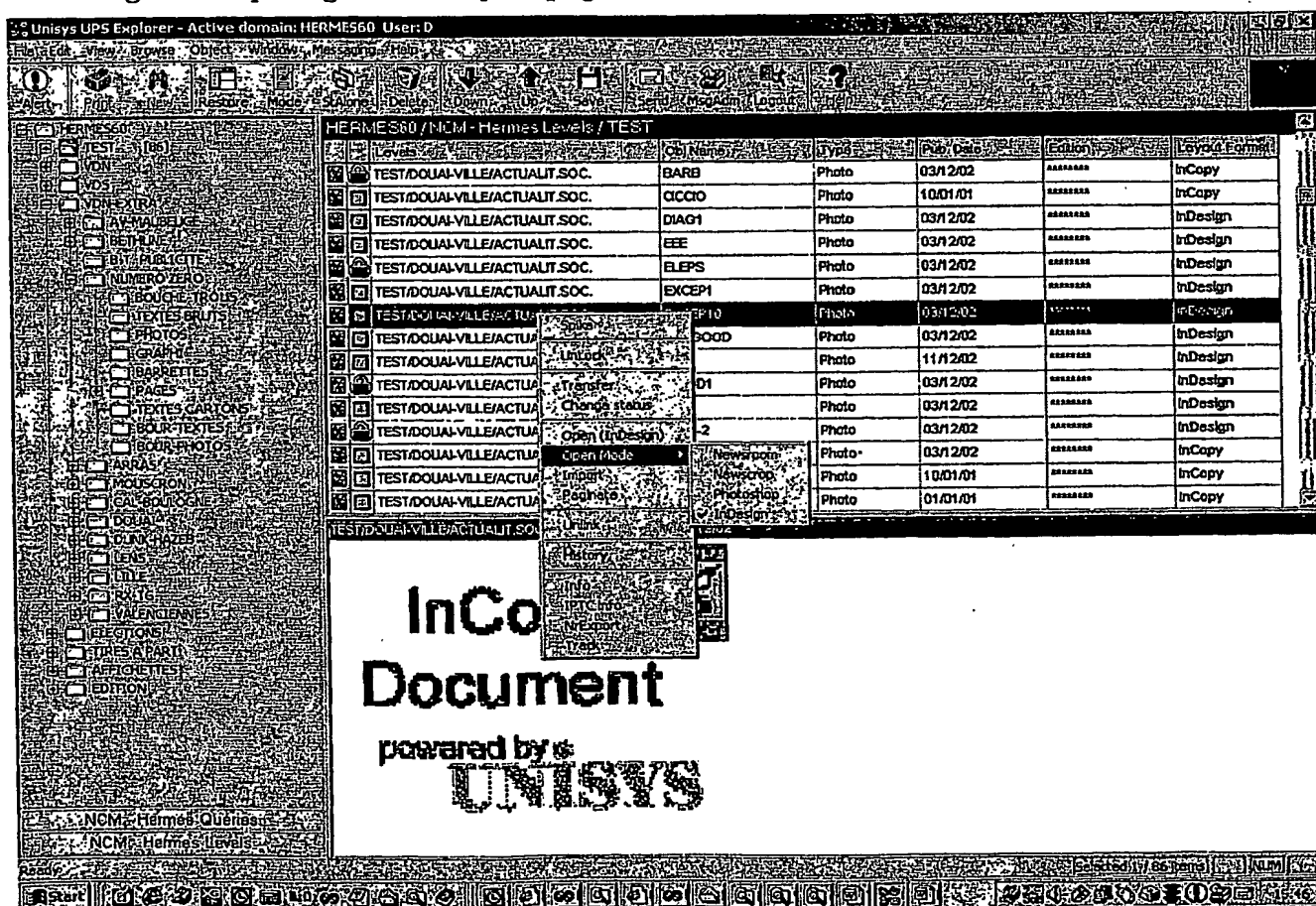
3.3 Opening retrieved external items

3.3.1 Description

Once retrieved from the Hermes database, the objects/pages created with third parties applications will be opened using the Hermes Explorer standard opening methods.

Users will be able to open the external object/page either by double clicking on the record displayed in the Hermes Explorer Object pane, or by right-clicking the selected record in the Browse pane. The latter option will display a popup menu that, through the **Open mode** menu item, will allow users to select the application the object/page must be open with. As an alternative, it will be possible to double click the selected object/page in the Browse window while pressing the Ctrl key. The object/page will be directly open in its native application.

Figure 4 –Opening external objects/pages



4 Information for User Manual

This functional specification, once approved, will be used by the technical writers to create the updates for the User Manual.

5 Performance

5.1 Speed

There will not be degradation in speed due to the implementation of this enhancement.

5.2 Reliability, Availability and Maintainability

There will not be degradation in reliability, availability and/or maintainability due to the implementation of this enhancement.

5.3 Capacity

There will not be degradation in capacity due to the implementation of this enhancement.

6 Installation

Not applicable, as this is an enhancement request to an existing product with established installation and trouble-shooting procedures.

This Page is inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ BLACK BORDERS
- ☒ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLORED OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REPERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images
problems checked, please do not report the
problems to the IFW Image Problem Mailbox**